

# Inverse Design of Curved Mechanical Metamaterials with Geometric AI

Mohammad Abu-Mualla, Jida Huang<sup>1</sup>

Department of Mechanical and Industrial Engineering, University of Illinois 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  $pos(e_m)$ ,  $pos(v_i)$ , and  $pos(v_j)$  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:

$$pos(e_m) = w_m pos(v_i) + (1 - w_m) pos(v_j)$$

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  $pos(f_m)$ ,  $pos(v_i)$ ,  $pos(v_j)$ , and  $pos(v_k)$  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:

$$pos(f_m) = w_{mi} pos(v_{mi}) + w_{mj} pos(v_{mj}) + w_{mk} pos(v_{mk})$$

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

---

<sup>1</sup> Corresponding author: jida@uic.edu



## 1.2. Property Calculations

For each generated structure, we computed two distinct sets of properties following homogenization under periodic boundary conditions. The first set comprises the linear effective stiffness properties, while the second set captures the non-linear compressive stress-strain response.

### 1.2.1. Effective Mechanical Properties Simulation

Numerical homogenization is a computational technique extensively employed to determine the homogenized macroscopic mechanical properties of cellular structures, with a particular focus on deriving the elasticity stiffness tensor [5]. The primary motivation for selecting the stiffness tensor as a representative metric for effective mechanical properties stems from its ability to capture all elastic constants via established approximation methods [6]. The numerical homogenization process involves solving the equation below, where  $E_{ijpq}$  denotes the stiffness tensor,  $\Omega$  represents the volume of the cellular structure, and  $\epsilon_{ij}$  corresponds to the strain field within the virtual displacement framework. Here,  $\epsilon_{pq}^{0(kl)}$  indicates the prescribed macroscopic strain, while  $v$  and  $\chi^{kl}$  denote the virtual and unknown displacement fields, respectively. For a comprehensive analysis of 3D cellular materials, the equation below must be solved under six independent loading scenarios, encompassing three axial and three shear deformation cases. A detailed methodological explanation can be found in reference [5].

$$\int_{\Omega} E_{ijpq} \epsilon_{ij}(v) \epsilon_{pq}(\chi^{kl}) d\Omega = \int_{\Omega} E_{ijpq} \epsilon_{ij}(v) \epsilon_{pq}^{0(kl)} dV \quad \forall v \in \Omega$$

Applying the homogenization approach results in a  $6 \times 6$  elastic stiffness tensor, denoted as  $[C_{ij}]$ . For subsequent analyses, the compliance tensor  $[S_{ij}] = [C_{ij}]^{-1}$  is used to derive the directional elastic constants for any 3D unit-cell configuration, as detailed in the equations below [6]:

$$\begin{aligned} E_1 &= \frac{1}{S_{11}}, & E_2 &= \frac{1}{S_{22}}, & E_3 &= \frac{1}{S_{33}}, \\ G_{23} &= \frac{1}{S_{44}}, & G_{13} &= \frac{1}{S_{55}}, & G_{12} &= \frac{1}{S_{66}}, \\ \nu_{23} &= -S_{23}E_2, & \nu_{13} &= -S_{13}E_1, & \nu_{12} &= -S_{12}E_1, \\ \nu_{32} &= -S_{23}E_3, & \nu_{31} &= -S_{13}E_3, & \nu_{21} &= -S_{12}E_2 \end{aligned}$$

This methodology offers a systematic framework for computing all relevant elastic moduli and Poisson's ratios, thereby enabling the comprehensive characterization of the mechanical properties for diverse unit-cell structures.

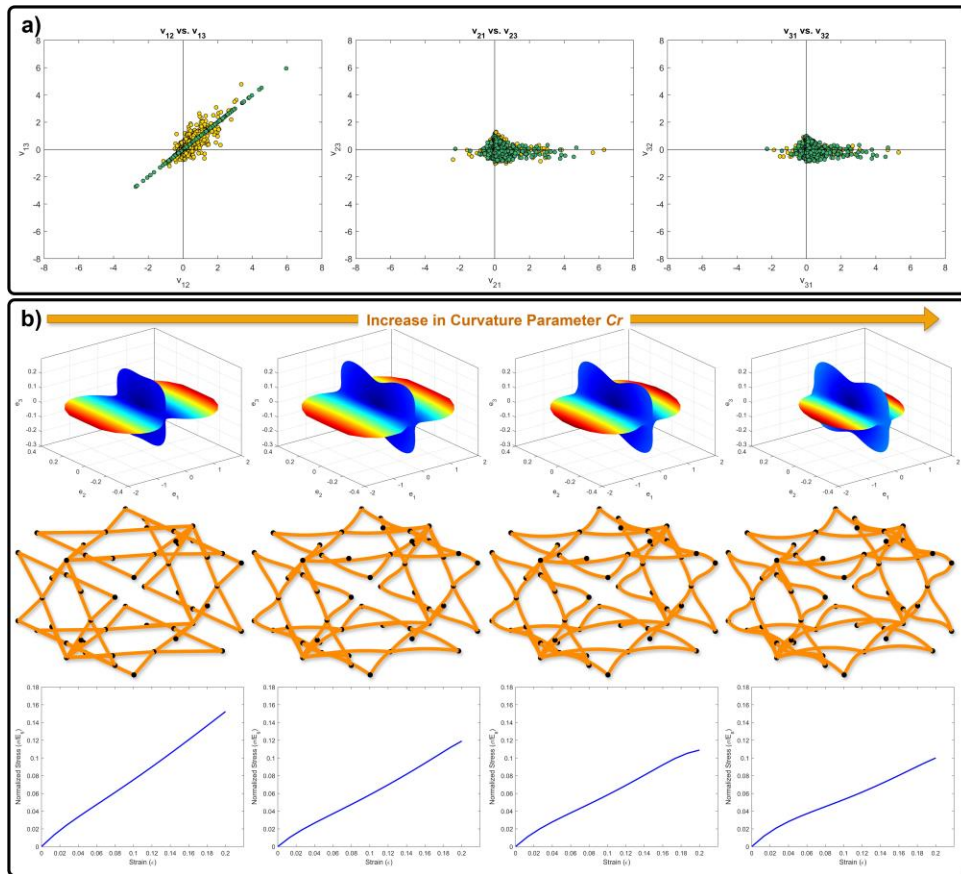
### 1.2.2. Compression Stress-Strain Simulation

To characterize the effective nonlinear response of our curved truss structures, we employed finite element analysis (FEA) using the open-source software FEBio [7]. The simulation framework was designed to extract the 3D stress-strain behavior of the unit-cells under uniaxial compression, leveraging principles of computational analysis with periodic boundary conditions applied to the displacements on the lateral four surfaces, while rotational constraints were enforced on all surfaces, thereby simulating the behavior of an infinitely periodic layer of truss unit-cells. The unit-cells were positioned between two rigid plates: a displacement boundary condition was applied at the top surface in the negative  $z$ -direction, while the bottom surface remained fixed. The truss structures were modeled using isotropic linear elastic material properties, defined by a Young's modulus  $E$  of 200 GPa and a Poisson's ratio  $\nu$  of 0.3. A compressive strain of up to 20%

was applied in incremental steps, with an implicit static solver used to ensure convergence, particularly under conditions of large deformation. The simulation employed 15 discrete time steps, each corresponding to a specific strain value, with adaptive control to address potential numerical challenges, such as localized buckling. The effective compressive stress was obtained by summing the reaction forces along the z-axis at the nodes in contact with the top plate. These reaction forces, combined with the applied displacements, were used to derive the stress-strain curve for each structure.

To enhance the efficiency of subsequent analyses, we selected representative stress values at specific strain increments (e.g.,  $\epsilon=1.33\%, 2.66\%, \dots, 20\%$ ) and reduced the dimensionality of the resulting data through interpolation. This streamlined representation of the compressive behavior ensures that critical mechanical properties, such as strain hardening and localized buckling, are captured efficiently for applications in soft robotics and energy-absorbing systems.

Supplementary Fig. 1a displays the Poisson's ratios ( $\nu_{12}$  vs.  $\nu_{13}$ ,  $\nu_{21}$  vs.  $\nu_{23}$ , and  $\nu_{31}$  vs.  $\nu_{32}$ ) 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:  $\nu_{12}$  vs.  $\nu_{13}$ ,  $\nu_{21}$  vs.  $\nu_{23}$ , and  $\nu_{31}$  vs.  $\nu_{32}$  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  $d_A = 40$ ,  $d_x = 40$ , and  $d_{Ax} = 50$ . Additionally, the reconstruction weight factor  $\lambda$  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 $P_\omega$
Input Dimension	210	25	90	90	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	90	90	210	25	37
Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU
Optimizer [8]	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 $p_\eta$
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 [8]	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 $P_\omega$	GPU*	5 hours 15 minutes
Training of the denoising network $p_\eta$	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.

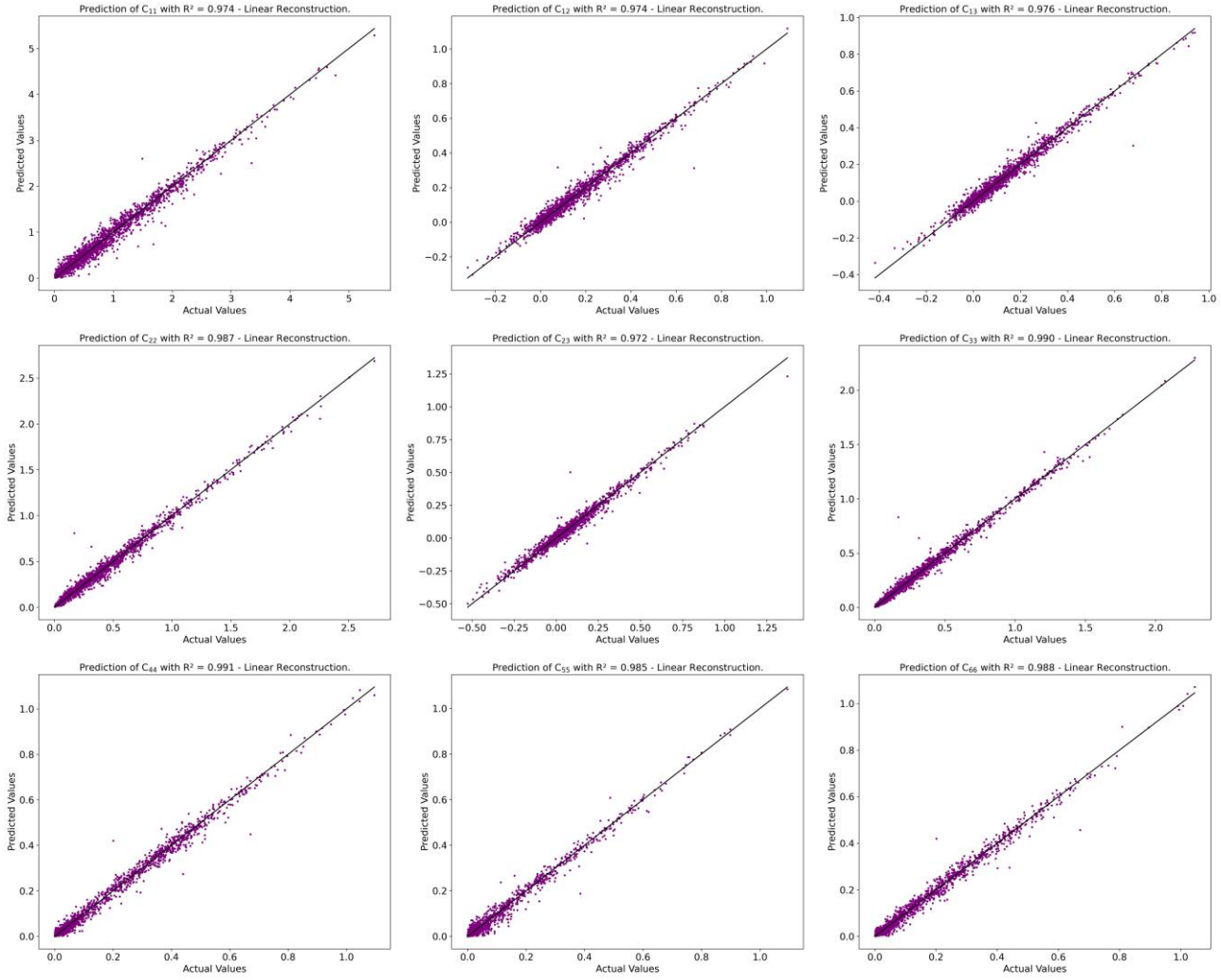
Supplementary Table 5: VAE reconstruction accuracy of the feature vector.

Column	R <sup>2</sup>	Column	R <sup>2</sup>	Column	R <sup>2</sup>	Column	R <sup>2</sup>	Column	R <sup>2</sup>
$w_0$	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 $P_\omega$ 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

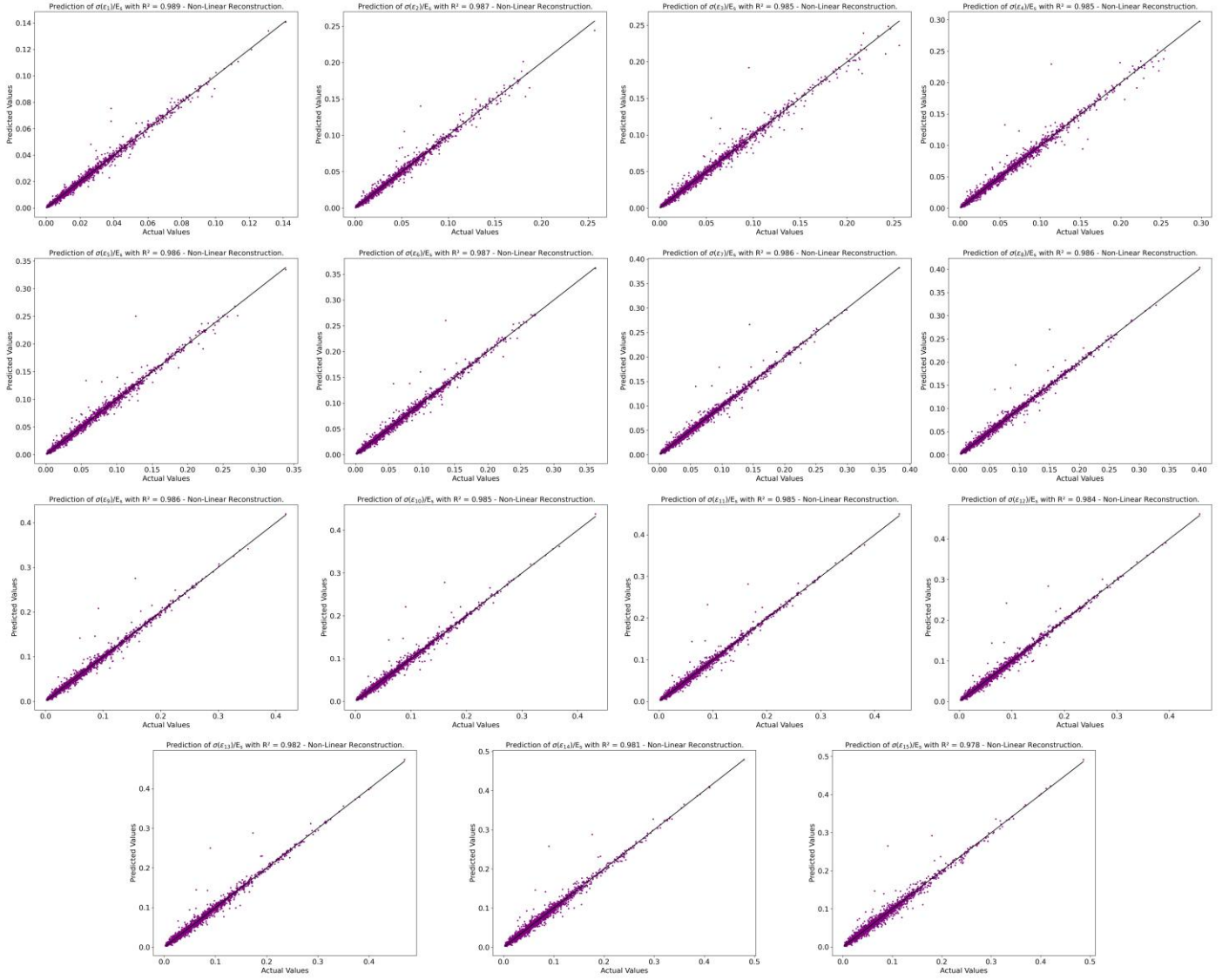
relevance, all subsequent figures focus exclusively on the prediction accuracy of the orthotropic stiffness components. The prediction performance of the linear stiffness tenor values is shown in Supplementary Fig. 2.



Supplementary Fig. 3: Assessment of the prediction accuracy of the property predictor model  $P_{\omega}$  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.

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.





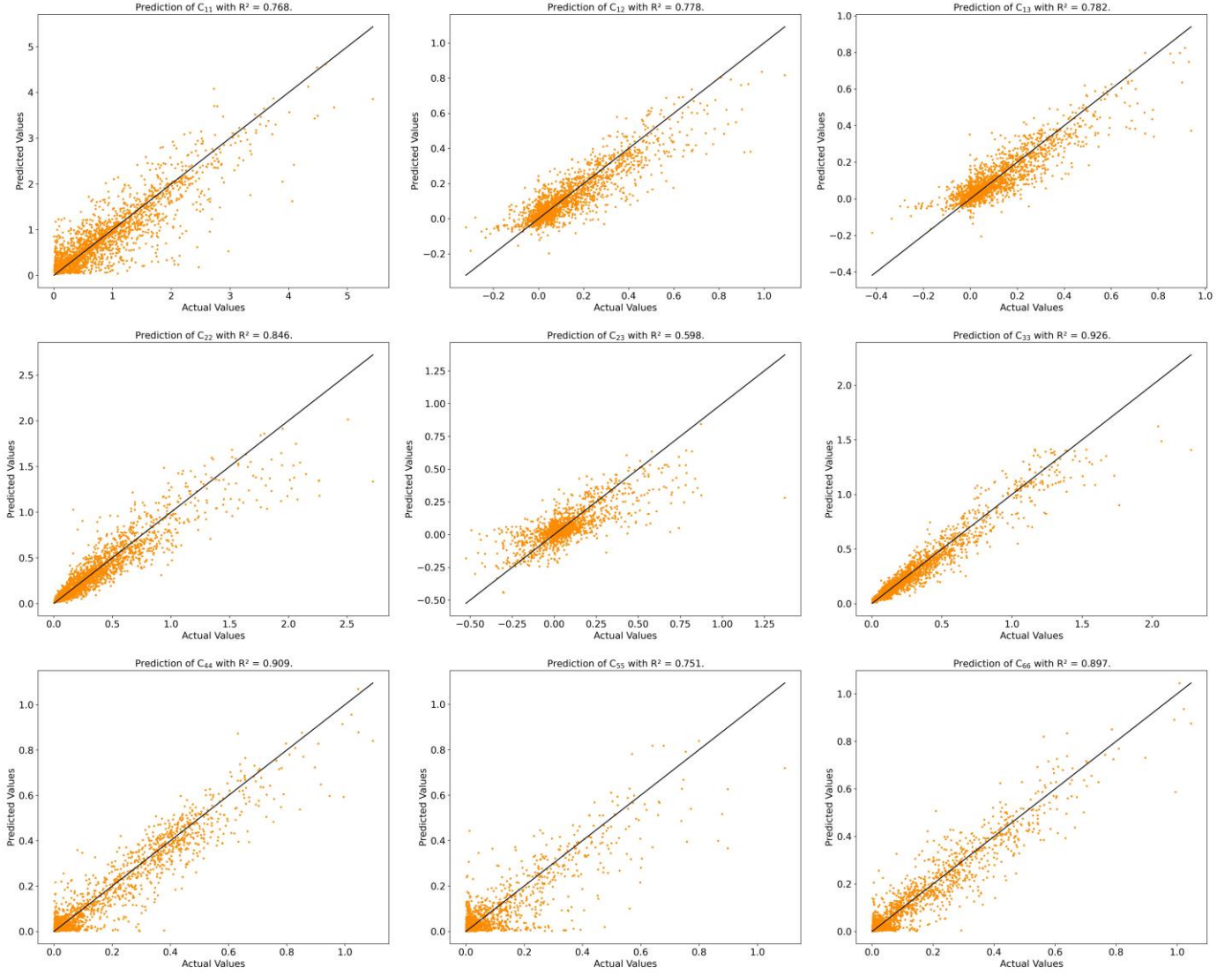
Supplementary Fig. 4: Assessment of the prediction accuracy of the property predictor model  $P_\omega$  in predicting nonlinear compression stress values  $\sigma(\epsilon)$ : A comparison between the predicted and true components of the compression stress values  $\sigma(\epsilon)$  in the test dataset.

### 2.2.3. Inverse Design Performance

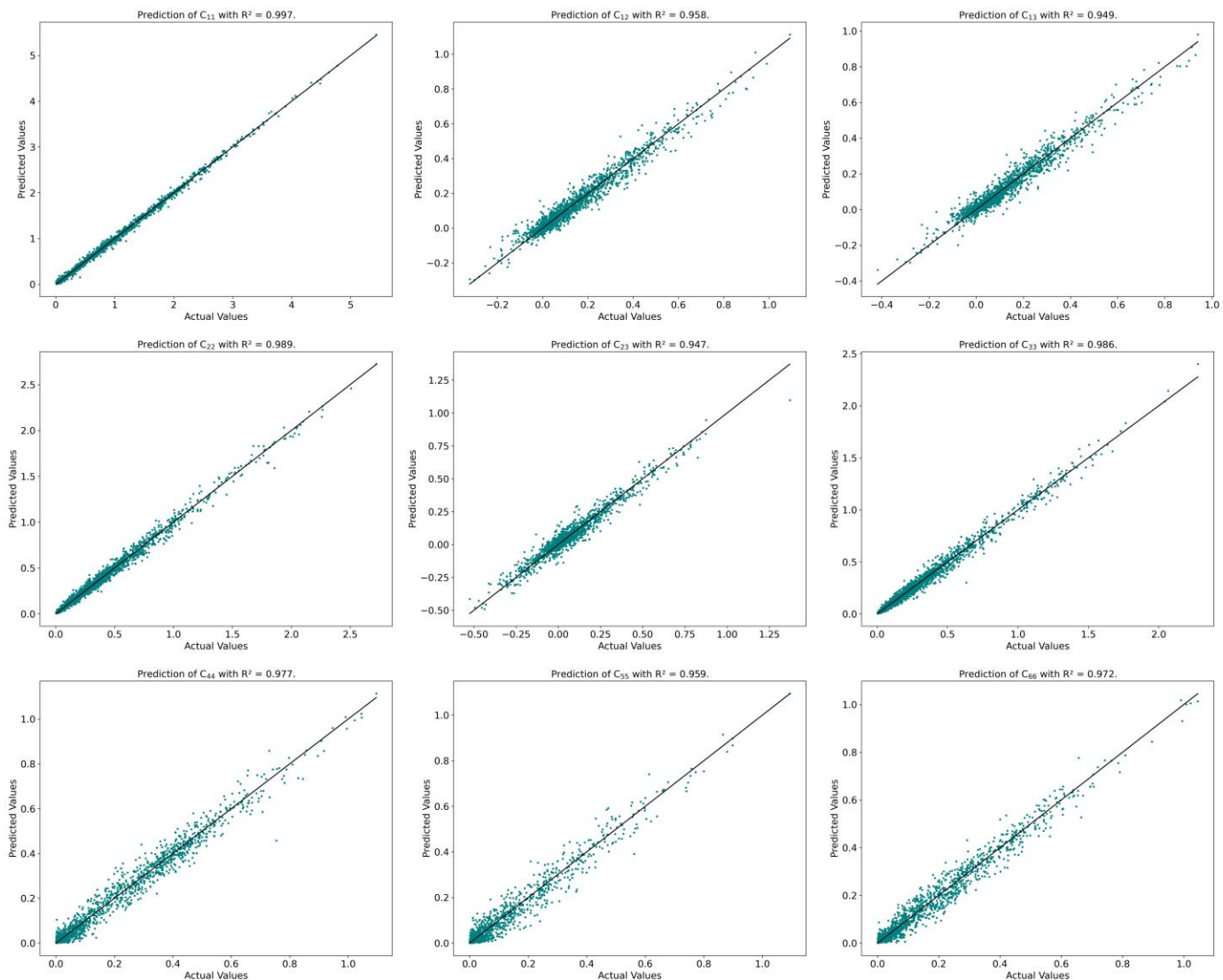
Building upon the analysis presented in Fig. 5 of the main manuscript, which demonstrated the inverse prediction accuracy for 2,000 unseen testing samples (with 50 samples generated for each testing point), we extend the evaluation to provide a more comprehensive insight into the prediction performance. 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  $R^2_{\text{Best}}$ . 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  $\mathbf{C}$ . Additionally, the mean prediction accuracy, represented as  $R^2_{\text{Mean}}$ , 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



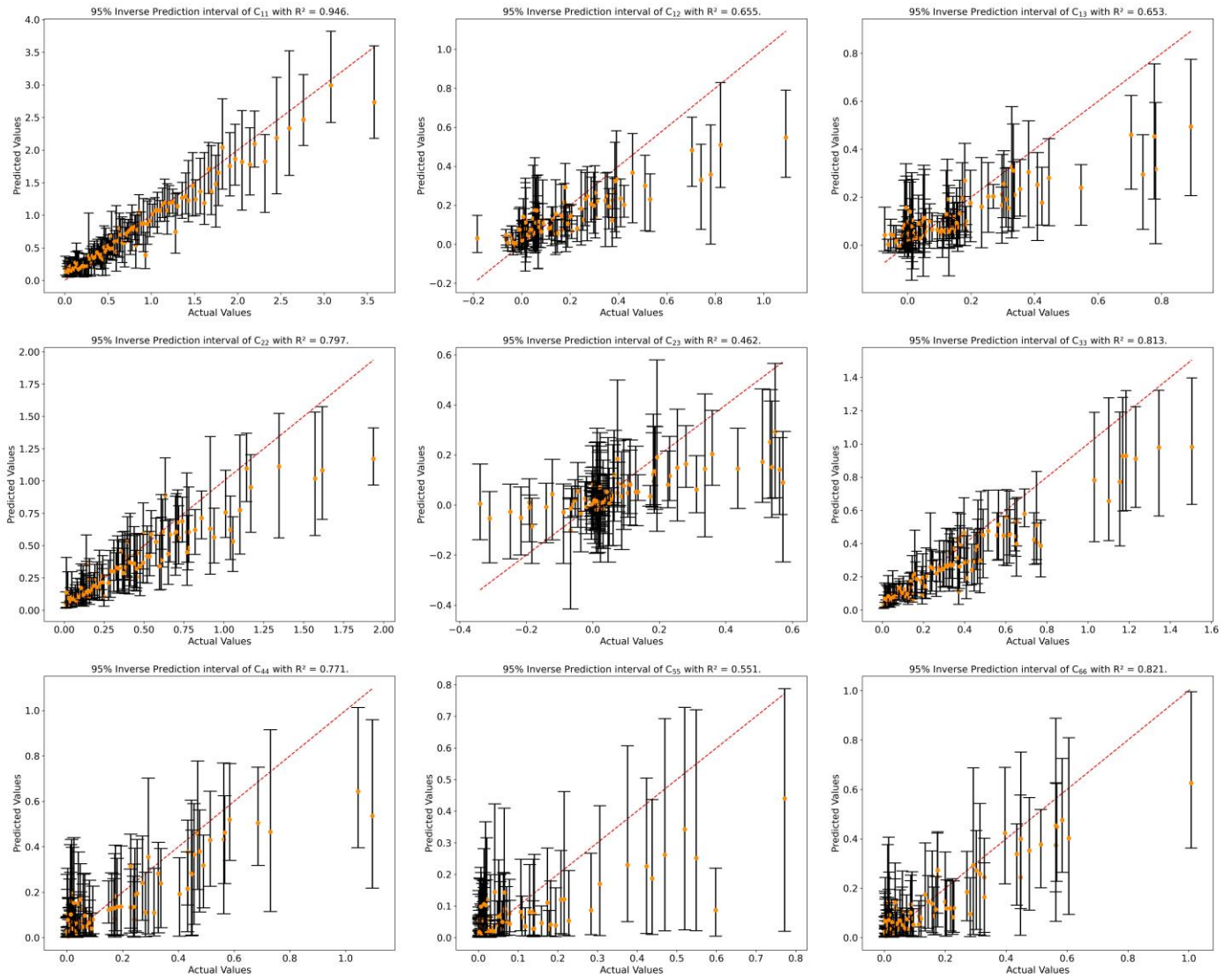
using the Wasserstein distance  $W$  relative to the target distribution, highlighting the distributional fidelity of the predictions for both the gradient-based optimization and diffusion model. This comprehensive analysis provides a nuanced understanding of the model's inverse prediction capabilities across various metrics and evaluation frameworks.



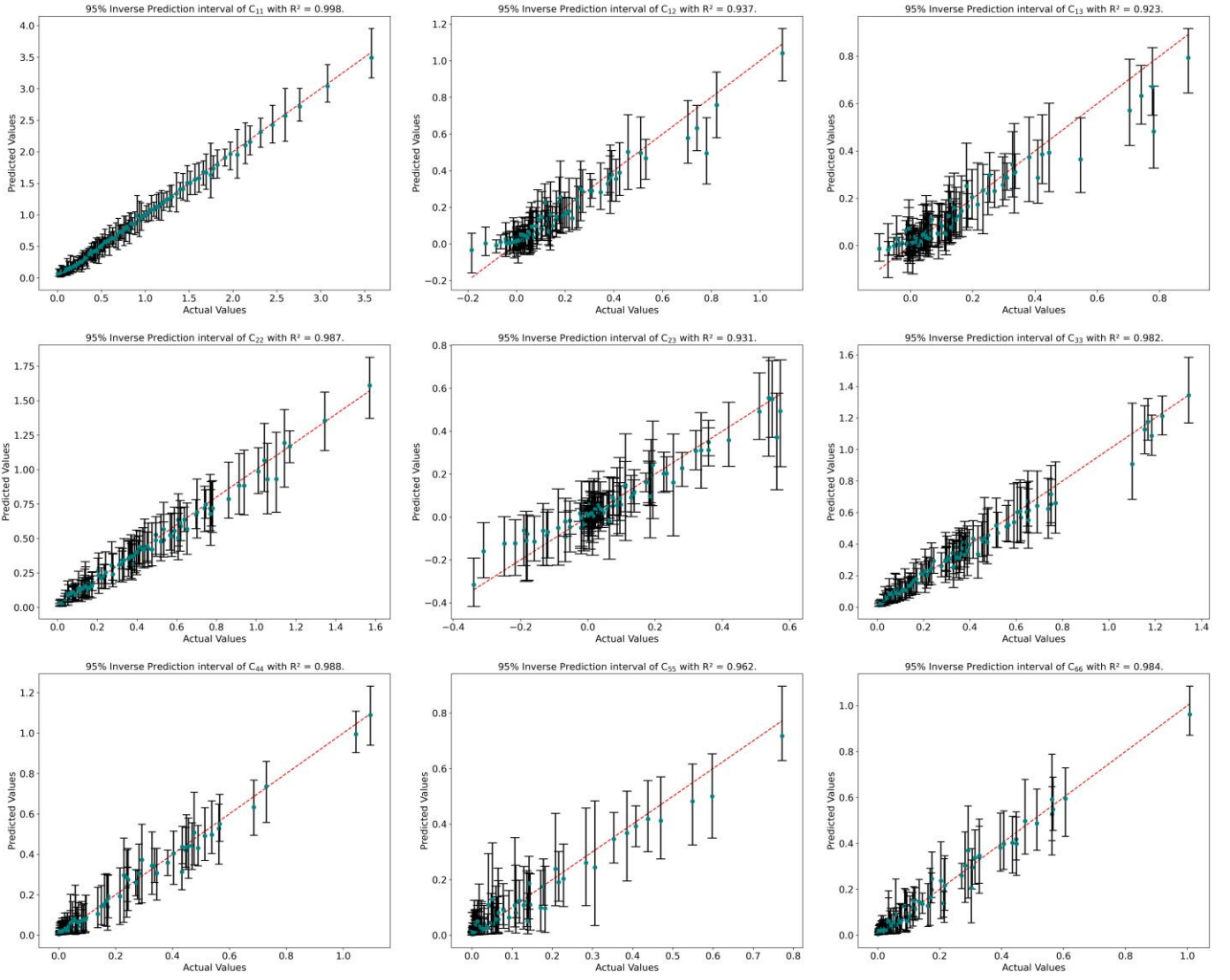
Supplementary Fig. 5: The inverse prediction accuracy, in terms of the sample with the lowest normalized mean square error (NMSE) [out of 50 reconstructed samples] relative to the target stiffness tensor  $\mathbf{C}$ , is quantified by the coefficient of determination  $R^2_{\text{Best}}$  for gradient-based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



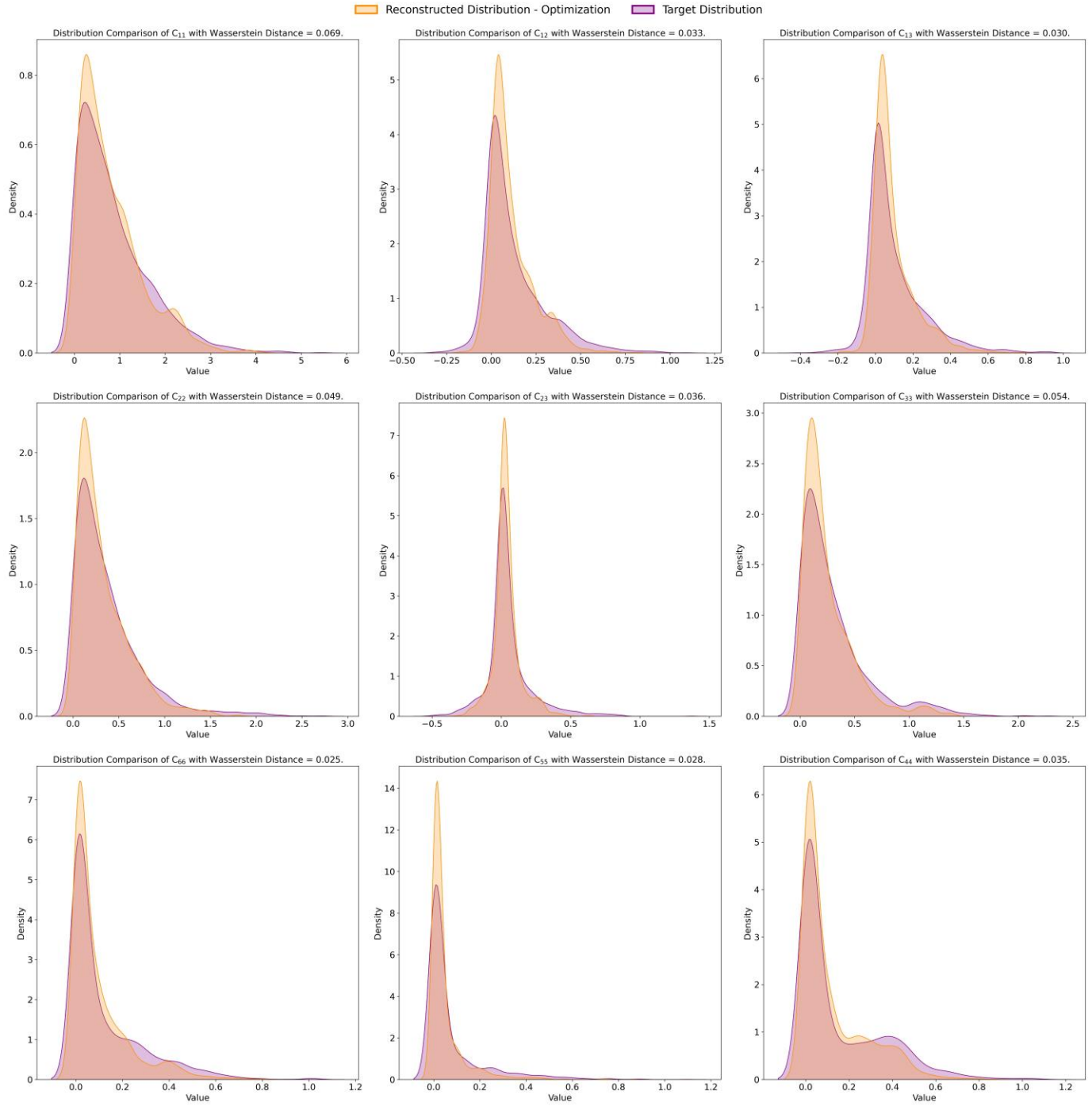
Supplementary Fig. 6: The inverse prediction accuracy, in terms of the sample with the lowest normalized mean square error (NMSE) [out of 50 reconstructed samples] relative to the target stiffness tensor  $\mathbf{C}$ , is quantified by the coefficient of determination  $R^2_{\text{Best}}$  for the latent diffusion model.



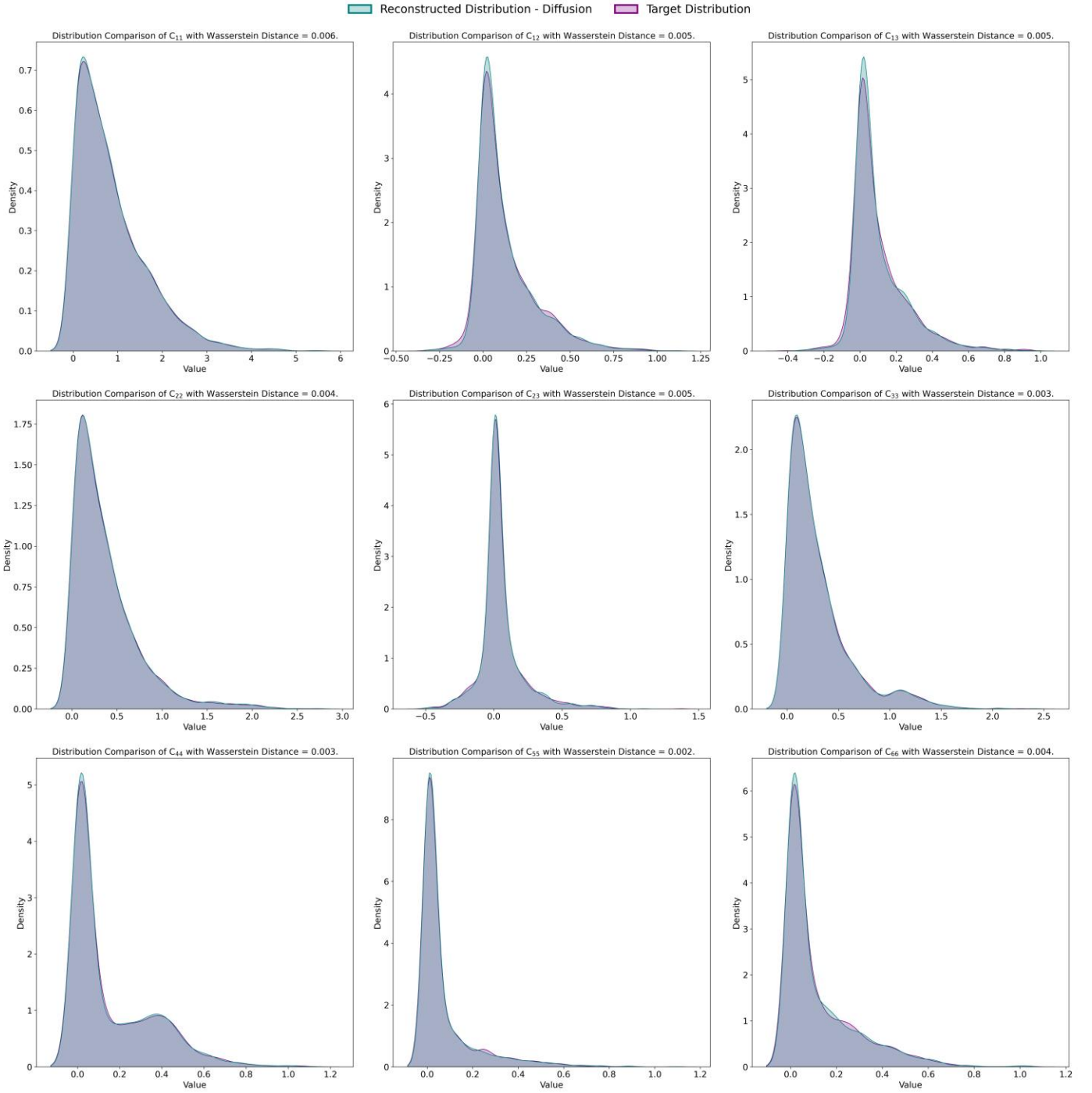
Supplementary Fig. 7: Inverse prediction accuracy with 95% confidence intervals for 100 unseen target stiffness tensors  $\mathbf{C}$ , based on 50 reconstructed samples. The mean prediction accuracy ( $R^2_{\text{Mean}}$ ) is calculated using the mean values of the reconstructed samples for each target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



Supplementary Fig. 8: 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. 9: 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.



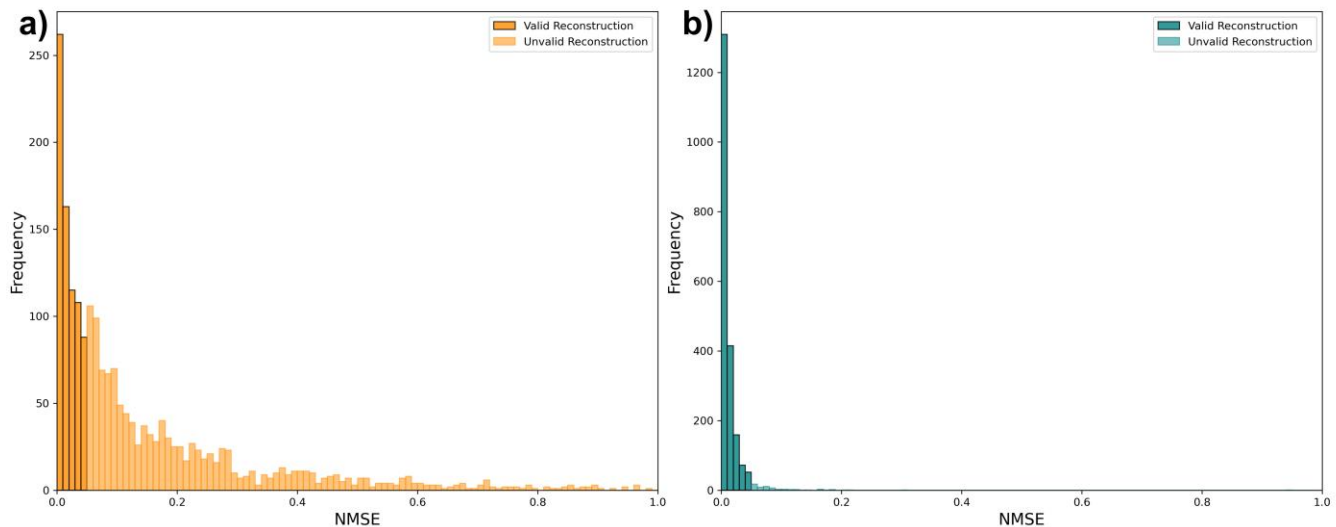
Supplementary Fig. 10: Distributions of reconstructed samples, evaluated using the Wasserstein distance  $W$  relative to the target stiffness tensor distributions, during predictions made with latent diffusion model.

We also compared the best reconstructed sample, out of 50, to the target stiffness tensor  $C$ . 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. S1 [9], was used as a quantitative metric to facilitate numerical comparison and evaluate the precision of the reconstructed stiffness tensor relative to the target.

$$\text{NMSE}(C, C) = \frac{\sum_{i=1}^6 \sum_{j=1}^6 (C_{\text{target},ij} - C_{\text{pred},ij})^2}{\sum_{i=1}^6 \sum_{j=1}^6 C_{\text{target},ij}^2} \quad (\text{S } 1)$$



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

## 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] Dong, Guoying, Yunlong Tang, and Yaoyao Fiona Zhao. "A 149 line homogenization code for three-dimensional cellular materials written in matlab." *Journal of Engineering Materials and Technology* 141, no. 1 (2019): 011005.
- [6] Lumpe, Thomas S., and Tino Stankovic. "Exploring the property space of periodic cellular structures based on crystal networks." *Proceedings of the National Academy of Sciences* 118, no. 7 (2021): e2003504118.
- [7] Maas, Steve A., Benjamin J. Ellis, Gerard A. Ateshian, and Jeffrey A. Weiss. "FEBio: finite elements for biomechanics." (2012): 011005.
- [8] Kingma, Diederik P. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).
- [9] Abu-Mualla, Mohammad, and Jida Huang. "Inverse design of 3D cellular materials with physics-guided machine learning." *Materials & Design* 232 (2023): 112103.