Supplementary information

Physics-informed neural networks for programmable origami metamaterials with controlled deployment

Sukheon Kang [a], Youngkwon Kim[b], Jinkyu Yang[b,*], and Seunghwa Ryu [a,c,*]

Affiliations

[a] Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

[b] Department of Mechanical Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

[c] KAIST InnoCORE PRISM-AI Center, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

[*]Corresponding author e-mail: ryush@kaist.ac.kr (S. Ryu)

[*]Corresponding author e-mail: jkyang11@snu.ac.kr (J. Yang)

Keywords

Kresling origami, origami, metamaterial, Bistable, Forward prediction, Inverse design, Physics-informed neural network, PINN

Section 1. Geometric and energetic modeling of conical Kresling origami

This section provides a detailed description of the geometric definitions, coordinate expressions, and energy formulation for the conical Kresling origami (CKO) structure. These formulations support the truss-based modeling introduced in Section 2.1 of the main manuscript and form the foundation for the physics-informed learning framework.

The CKO structure consists of n identical unit cells arranged in a periodic manner. Each cell is defined by five geometric parameters: the top and bottom edge lengths a and b, the mountain and valley crease lengths c and d, and the internal angle β between the bottom edge and its adjacent mountain crease. The folded structure forms a conical shell in which the top and bottom polygons are inscribed in circles of radius r and R, respectively, given by:

$$r = \frac{a}{2\sin(\frac{\pi}{n})} \tag{S1}$$

$$R = \frac{b}{2\sin(\frac{\pi}{n})} \tag{S2}$$

where n is the number of unit cells. The folding configuration is described by two state variables: the vertical height h between the polygons and the relative rotational angle φ between them. The spatial coordinates of the vertices are then defined in cylindrical coordinates as follows, where $i = 1,2,\dots,n$

$$A_i = \begin{bmatrix} r\cos(\frac{2\pi(i-1)}{n} + \varphi) \\ r\sin(\frac{2\pi(i-1)}{n} + \varphi) \\ h \end{bmatrix} \tag{S3}$$

$$B_i = \begin{bmatrix} Rcos(\frac{2\pi(i-1)}{n}) \\ Rsin(\frac{2\pi(i-1)}{n}) \\ 0 \end{bmatrix} \quad \text{(S4)}$$

Using these coordinates, the deformed lengths of the mountain and valley creases, denoted $\tilde{c}(h,\varphi)$ and $\tilde{d}(h,\varphi)$, respectively, can be computed as the Euclidean distances between corresponding vertices, where c and d represent the initial lengths of the mountain and valley creases.

$$\tilde{c}(h,\varphi) = |A_i - B_i| = \sqrt{h^2 + r^2 + R^2 - 2rRcos(\varphi)} \quad \text{(S5)}$$

$$\tilde{d}(h,\varphi) = |A_{i+1} - B_i| = \sqrt{h^2 + r^2 + R^2 - 2rRcos(\varphi + \frac{2\pi}{n})} \quad \text{(S6)}$$

These expressions describe the nonlinear coupling between the vertical displacement and rotational deformation of the CKO structure.

Using the deformed crease lengths, the total strain energy of the CKO structure can be expressed as the sum of the elastic strain energies stored in all mountain and valley creases:

$$U(h,\varphi) = \sum_{i=1}^{n} [\frac{nk_m}{2}(\tilde{c}(h,\varphi) - c)^2 + \frac{nk_v}{2}(\tilde{d}(h,\varphi) - d)^2] \quad \text{(S7)}$$

where $k_m = \frac{EA}{c}$ and $k_v = \frac{EA}{d}$ denote the axial stiffness of the mountain and valley creases, respectively. Here, E is the Young's modulus and A is the cross-sectional area of the crease elements. This energy expression provides a foundation for evaluating equilibrium states, stiffness, and energy barriers between configurations.

The equilibrium states of the structure for a given vertical height h correspond to torque-free configurations, obtained by enforcing:

$$\frac{\partial \mathrm{U}}{\partial \varphi} = 0 \tag{S8}$$

Solving Eq. (S8) yields the equilibrium rotation profile $\varphi(h)$, which can then be substituted back into Eq. (S7) to compute the minimum-energy path. This minimum-energy trajectory corresponds to the red dashed curves shown in Figure 1(D) of the main text and forms the basis for the energy–height profiles illustrated in Figure 1(E).

These formulations fully describe the geometric and energetic behavior of the CKO structure within the truss-based modeling framework. They provide the physically consistent, data-free foundation for the physics-informed neural network (PINN) approach used in the forward prediction and inverse design tasks described in the main manuscript.

Section 2. Algorithms and Implementation Details

This section provides detailed algorithms and implementation specifications for the physics-informed neural network (PINN) framework used in forward prediction and inverse design of conical Kresling origami structures. Three algorithms are presented: Algorithm 1 for forward prediction, Algorithm 2 for inverse design based on target energy curves, and Algorithm 3 for inverse design with bistable energy programming.

2.1 Network Architecture and Implementation

We employed different network architectures for forward prediction and inverse design problems. All networks were implemented in PyTorch. For forward prediction (Algorithm 1), we utilized a fully connected neural network consisting of two hidden layers with 128 neurons each, activated by hyperbolic tangent functions. The input layer receives only the sampled height vector $\boldsymbol{h}$, while the geometric parameters (a, b, c, β, n) are treated as fixed constants inside the energy model. The output layer employs a sigmoid-based transformation to enforce physical constraints on the rotation angle φ, mapping it to the admissible range $[\varphi_{lower}, \varphi_{upper}]$, where $\varphi_{upper} = min(\varphi_f, \varphi_s)$ represents the minimum of the flat-folding and locked configuration angles.

For inverse design problems (Algorithms 2 and 3), the framework comprises two coupled models: a PhysicsNN for predicting the rotation field and a ParameterModel for optimizing structural parameters. The PhysicsNN maintains the same architecture as the forward prediction network with two hidden layers of 128 neurons each. The ParameterModel employs parameter-specific transformations to ensure physical validity: logarithmic transformations for positive parameters a and b, and sigmoid transformations for constrained parameters $c \in [|r - R|, r + R]$ and $\beta \in (0, \pi)$.

2.2 Loss Function Components and Their Roles

The loss functions in our framework consist of multiple components, each serving specific purposes:

1. Energy Minimization Loss (L_E): Drives the network to minimize the total potential energy, ensuring physical accuracy.
2. Euler-Lagrange Residual Loss (L_{EL}): Promotes smoothness in the predicted rotation profile $\varphi(h)$ and prevents abrupt curvature changes or nonphysical jumps in the resulting energy curve.
3. Physics Equilibrium Loss (L_{phys}): Minimizes rotational torque ($\partial \mathrm{U} / \partial \varphi = 0$) and ensures static equilibrium across the deformation path.
4. Target Matching Loss (L_{target}): Enforces consistency between the predicted energy curve and the given target profile.
5. Barrier Constraint Loss ($L_{barrier}$): Ensures that the energy gradient vanishes at the inferred barrier point ($L_{barrier,1}$) and that the barrier is a unique maximum through convexity checking ($L_{barrier,2}$).
6. Smoothness Loss (L_{smooth}): Regularizes the energy curve globally, preventing sharp curvature transitions.
7. Constraint Penalty ($L_{penalty}$): A hard constraint enforcement mechanism that applies a large penalty ($\mathrm{C} = 10^3$) when geometric or physical feasibility constraints are violated.

2.3 Training Hyperparameters

The training hyperparameters were carefully selected based on the complexity of each task. For forward prediction (Algorithm 1), we employed the Adam optimizer with a learning rate of 1e-3. The loss function incorporates an Euler-Lagrange regularization term with weight $\lambda_{EL} = 0.1$ and a smoothness coefficient $\lambda_{reg} = 0.01$.

For inverse design based on full energy curves (Algorithm 2), we used a learning rate scheduler (ReduceLROnPlateau) with a reduction factor of 0.5 and patience of 100 epochs. The multi-objective loss function was weighted as follows: $\lambda_{target} = 20.0$ for target energy matching, $\lambda_{phys} = 0.1$ for physics equilibrium constraints, and $\lambda_{EL} = 0.1$ for variational consistency.

For bistable energy programming (Algorithm 3), additional loss terms were introduced to control the energy landscape. The weights were set as $\lambda_{target} = 10.0$, $\lambda_{phys} = 1.0$, $\lambda_{EL} = 0.1$, $\lambda_{smooth} = 0.05$ for global smoothness, and $\lambda_{barrier} = 5.0$ for barrier energy constraints.

Each case was trained for up to 50 k epochs on a CPU workstation (Intel i9 processor, 64 GB RAM) without GPU acceleration. All computations were performed using single-precision floating-point arithmetic (float32) to balance computational efficiency with numerical accuracy. Physical constants were normalized with E = A = 1 for dimensional consistency. Constraint violations were penalized with a large constant C = 1000 to effectively restrict the parameter space to physically realizable configurations.

2.4 Algorithms

Algorithm 1 Forward design of origami structure using PINN

Input: Structural parameters (a, b, c, β, n), domain $[h_{min},\ h_{max}]$

Output: Energy curve $U(h)$, rotation angle $\varphi(h)$

Initialize PINN model to predict $\varphi(h)$

Compute physical constraints: $\varphi_{lower} = 0,\ \varphi_{upper} = \min(\varphi_f, \varphi_s)$

repeat

    Predict $\varphi(h)$ using PINN

    Compute energy $U(h, \varphi(h))$ using truss model

    Compute physical loss: $\partial U/\partial \varphi = 0$

    Compute 1st derivative: $\partial \varphi/\partial h$

    Compute 2nd derivative: $\partial^2 \varphi/\partial h^2$

    Compute EL residual: $\partial U/\partial \varphi - 2\lambda_{reg}(\partial^2 \varphi/\partial h^2)$

    Compute EL loss: $L_{EL} = \text{mean}[(\partial U/\partial \varphi - 2\lambda_{reg}(\partial^2 \varphi/\partial h^2))^2]$

    Compute total loss: $L_{total} = L_E + \lambda_{EL} L_{EL}$

    Update PINN parameters using gradient descent

until Max epoch is reached;

Return $U(h, \varphi(h)),\ \varphi(h)$

Algorithm 2 Inverse design of origami structure based on target energy curve using PINN

Input: Initial structural parameters (a, b, c, β, n), Target energy curve $U_{target}(h)$

Output: Optimized structural parameters (a, b, c, β)

1: Initialize parameter model and physics model (PINN)

2: **repeat**

3: Get current structural parameter (a, b, c, β) from parameters model

4: Check physical validity of parameters

5: Predict $\varphi(h)$ using physics model

6: Compute energy $U(h, \varphi(h))$ using truss model

7: Compute target matching loss: $L_{target} = \text{mean}[(U(h) - U_{target}(h))^2]$

8: Compute physical loss: $L_{phys} = \text{mean}[(\partial U/\partial \varphi)^2]$

9: Compute EL loss as in Algorithm 1

10: Compute total loss: $L_{total} = \lambda_{target} L_{target} + \lambda_{phys} L_{phys} + \lambda_{EL} L_{EL} + L_{penalty}$

11: Update PINN parameters using gradient descent

12: **until** Max epoch is reached;

13: Return optimized structural parameters (a, b, c, β)

Algorithm 3 Inverse design of origami structure with bistable energy programming using PINN

Input: Initial structural parameters (a, b, c, β, n), Target energy barrier value $U_{barrier}$,
Two stable points $(h_1, U_1 = 0), (h_2, U_2 = 0)$

Output: Optimized structural parameters (a, b, c, β)

Initialize parameter model and physics model (PINN)

Find indices of stable points h_1 and h_2 in discretized domain

Define barrier search region: $[min(h_1, h_2), max(h_1, h_2)]$

repeat

  Get current structural parameters (a, b, c, β) from parameter model

  Check physical validity of parameters

  Predict $\varphi(h)$ using physics model

  Compute energy $U(h, \varphi(h))$ using truss model

  Extract energy values at stable points

  Compute stable points matching loss: $L_{target} = \text{mean}[(U(h_i) - U_{target}(h_i))^2]$ for $i \in \{1,2\}$

  Compute physical loss: $L_{phys} = \text{mean}[(\partial U/\partial \varphi)^2]$

  Compute EL loss as in Algorithm 1

  Compute smoothness loss: $L_{smooth} = \text{mean}[(\partial^2 U/\partial \varphi^2)^2]$

  Find energy maximum in barrier search region: $h_{max} = argmax_{h \in [h_1, h_2]} U(h)$

  Compute barrier gradient constraint: $L_{barrier,1} = \left|\frac{\partial U}{\partial h}\right|^2_{h=h_{max}}$

  Compute barrier convexity constraint: $L_{barrier,2} = \text{ReLU}(\frac{\partial^2 U}{\partial h^2})_{h=h_{max}}$

  Compute total barrier loss: $L_{barrier} = L_{barrier,1} + L_{barrier,2}$

  Compute total loss: $L_{total} = \lambda_{target} L_{target} + \lambda_{phys} L_{phys} + \lambda_{EL} L_{EL} + \lambda_{smooth} L_{smooth} + \lambda_{barrier} L_{barrier} + L_{penalty}$

  Update PINN and parameter model using gradient descent

until Max epoch is reached;

Return optimized structural parameters (a, b, c, β)

Section 3. Inverse design based on full energy curves - Detailed parameters

This section provides detailed structural parameters and accuracy metrics for the inverse design framework based on full energy curves (Section 3.2 of the main manuscript). The parameters correspond to the four representative cases shown in Figure 5.

Case	Parameter [mm]	Target	Inverse PINN
Case 1	a	1	1.3210
	b	2	1.6450
	c	3.1	2.9653
	β [rad]	1.5	1.6542
Case 2	a	1	1.3977
	b	2	1.5078
	c	3.5	3.1377
	β [rad]	1.5	1.7253
Case 3	a	1	1.3773
	b	2	1.5339
	c	3.3	3.0579
	β [rad]	1.5	1.6962
Case 4	a	1	1.3611
	b	2	1.5427
	c	3.5	3.2718
	β [rad]	1.5	1.6748

Supplementary Table S1. Structural parameters for inverse design of CKO geometries. Target parameters used to generate reference energy curves and inverse-designed parameters obtained through the PINN framework for cases in Figure 5.

Case	R^2 (Target)	RMSE (Target)	R^2 (FEA)	RMSE (FEA)
Case 1	0.993539	0.000097	0.998506	0.000045
Case 2	0.999589	0.000321	0.991991	0.001347
Case 3	0.999947	0.000039	0.997255	0.000270
Case 4	0.999925	0.000047	0.998268	0.000217

Supplementary Table S2. Accuracy metrics for inverse design based on full energy curves. R^2 and RMSE values comparing PINN predictions with target energy curves and FEA validation results for the four cases in Figure 5.

Section 4. Additional results for inverse design with bistable energy programming

This section presents comprehensive validation results for the inverse design framework with bistable energy programming across seven different design scenarios. We systematically varied the second stable point height (h_2), target energy barrier values ($U_{barrier}$), and the number of polygon units (n) to demonstrate the robustness and versatility of our approach. All figures and tables in this section are arranged in the following order: (1) h_2 = 3.4 mm, $U_{barrier}$ = 0.05 mJ; (2) h_2 = 3.9 mm, $U_{barrier}$ = 0.05 mJ; (3) h_2 = 4.4 mm, $U_{barrier}$ = 0.05 mJ; (4) h_2 = 3.9 mm, $U_{barrier}$ = 0.075 mJ; (5) h_2 = 3.9 mm, $U_{barrier}$ = 0.1 mJ; (6) h_2 = 3.9 mm, $U_{barrier}$ = 0.075 mJ with n = 9; (7) h_2 = 3.9 mm, $U_{barrier}$ = 0.075 mJ with n = 10. Unless otherwise specified, n = 8 polygon units were used.

Supplementary Figure 1 shows the inverse-designed energy curves for all seven cases, each validated against finite element analysis (FEA) simulations. The first three cases (h_2 = 3.4, 3.9, and 4.4 mm) maintain a constant barrier height of 0.05 mJ while varying the second stable point location, demonstrating the framework's ability to control deployment distance. The next two cases explore different barrier heights (0.075 and 0.1 mJ) at a fixed h_2 = 3.9 mm, validating precise control over deployment forces. The final two cases examine the effect of polygon number (n = 9 and 10) while maintaining identical energy specifications, confirming the framework's adaptability to different geometric configurations.

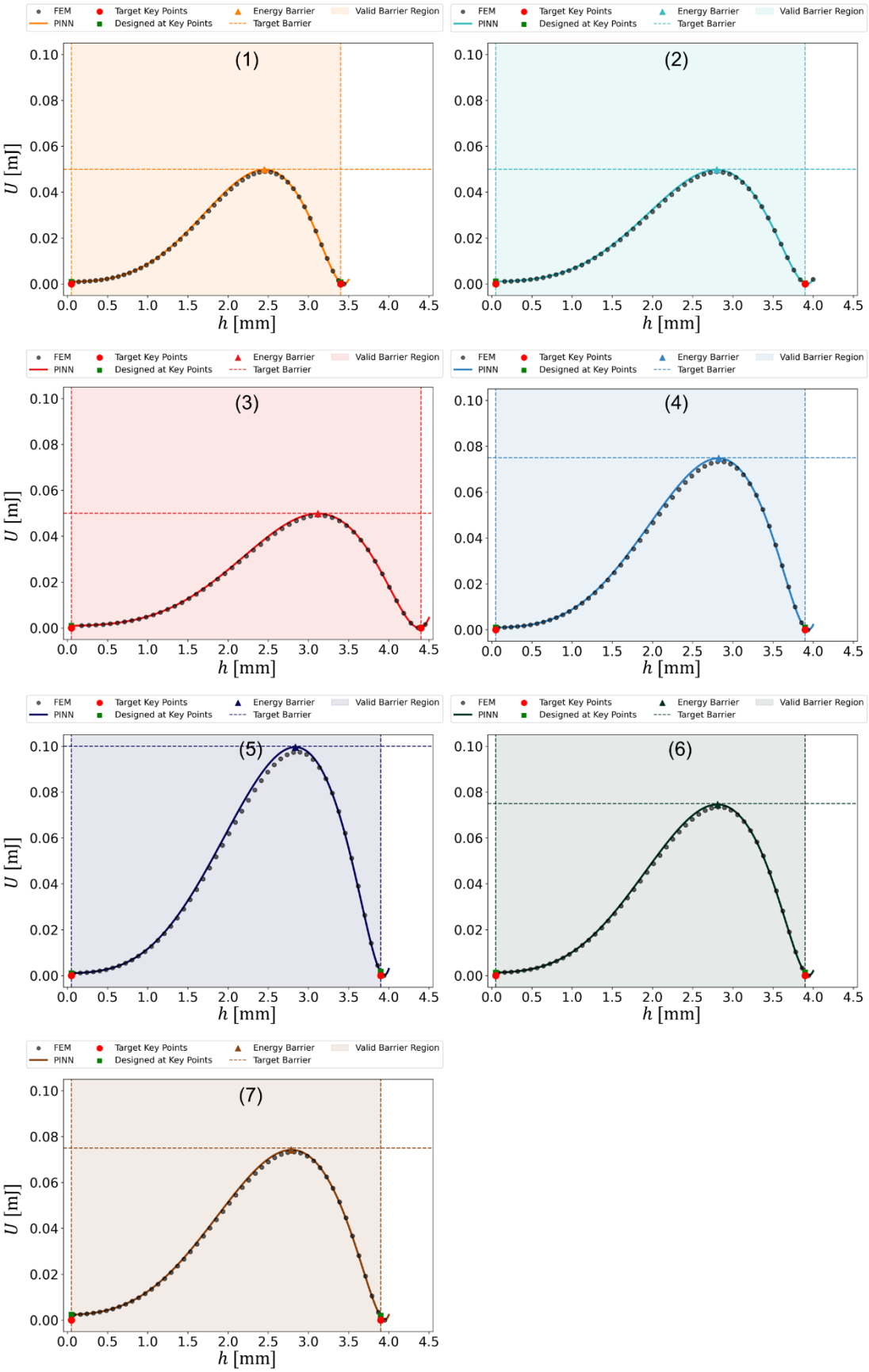
The training convergence behavior is illustrated in Supplementary Figure 2, which displays the evolution of total loss and its components across all cases. All designs achieved stable convergence within 50,000 epochs, with the total loss typically reaching values below 10^{-4}. The loss curves exhibit characteristic two-phase behavior: rapid initial descent followed by gradual refinement, indicating robust optimization dynamics. Notably, cases with higher polygon numbers (n = 9 and 10) demonstrated slower loss reduction compared to the n = 8

cases, suggesting that the increased geometric complexity requires more careful navigation of the optimization landscape to achieve the desired energy specifications.

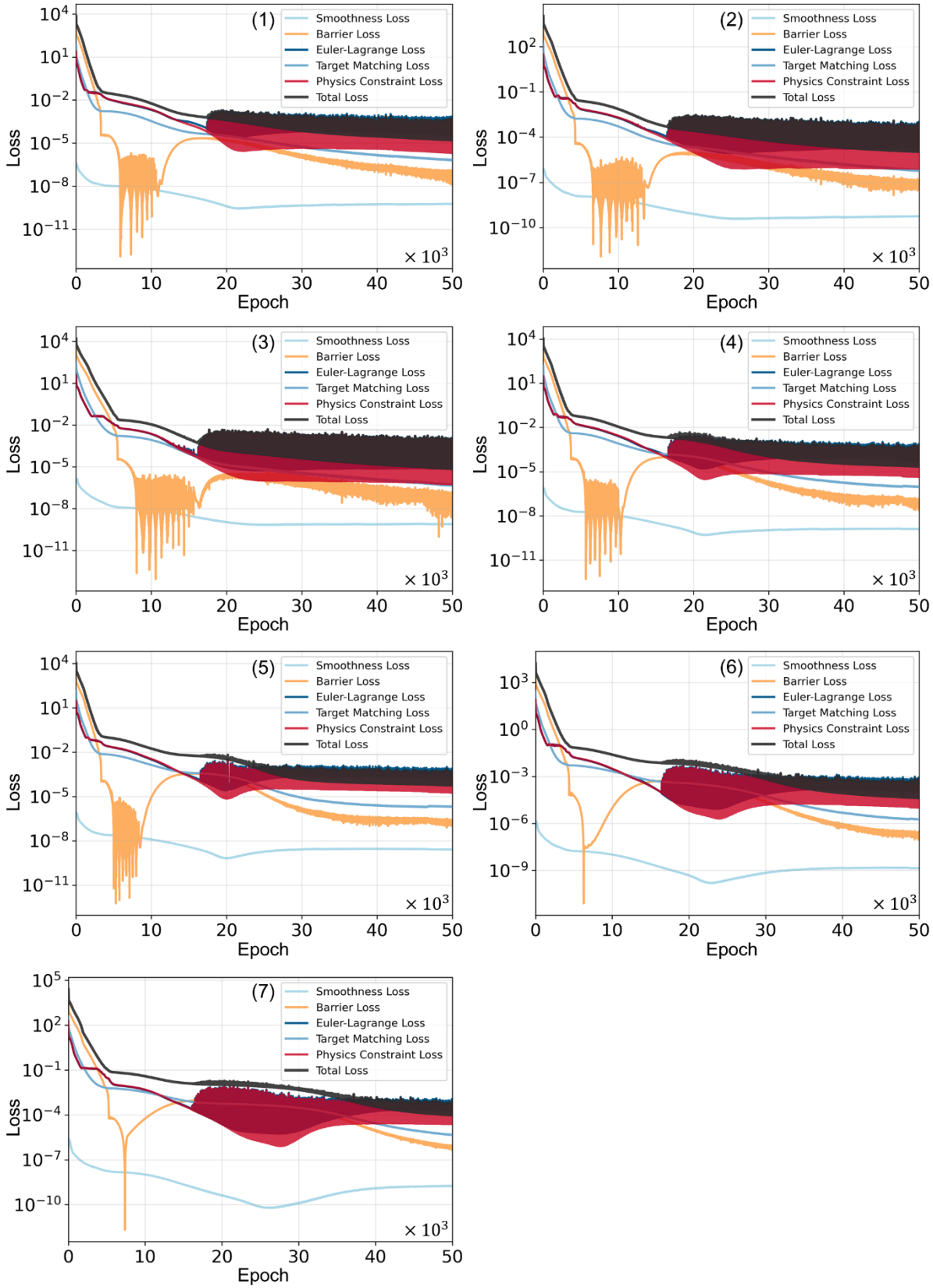
Supplementary Figure 3 presents the evolution of structural parameters (a, b, c, β) during training for each case. The parameter trajectories demonstrate smooth convergence without oscillations, validating the effectiveness of our parameter transformation strategies. Interestingly, designs with different polygon numbers (n = 8, 9, 10) converged to distinct parameter sets while achieving identical energy specifications, highlighting the non-unique nature of the inverse design problem and the framework's ability to find valid solutions within the feasible design space.

Quantitative validation results are summarized in Supplementary Table 1. The agreement between PINN predictions and FEA simulations is excellent across all cases, with coefficients of determination (R^2) exceeding 0.998 and root mean square errors (RMSE) below 0.0016. The highest accuracy was achieved for the h_2 = 4.4 mm case (R^2 = 0.9993, RMSE = 0.000469), while the largest RMSE occurred for the highest barrier case ($U_{barrier}$= 0.1 mJ), which can be attributed to the increased nonlinearity in the energy landscape. These metrics confirm that the physics-informed learning successfully captures the complex mechanical behavior across diverse design specifications.

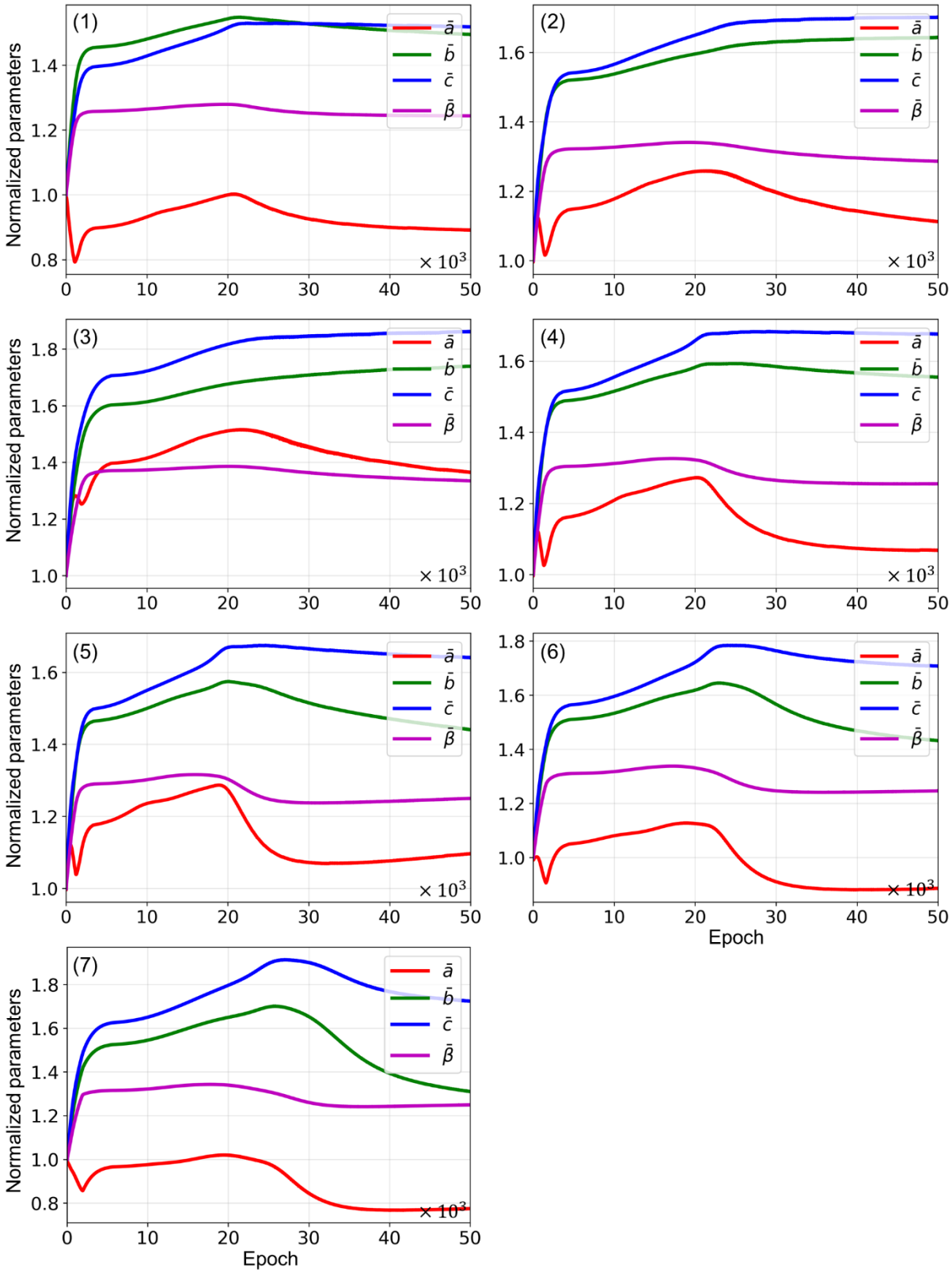
The consistency of results across varying geometric configurations (n = 8, 9, 10) with identical energy targets demonstrates that our framework can accommodate different structural realizations of the same mechanical behavior. This flexibility is particularly valuable for practical applications where manufacturing constraints or integration requirements may dictate specific geometric configurations. Furthermore, the successful inverse design across a wide range of barrier heights (0.05 to 0.1 mJ) validates the framework's applicability to diverse deployment force requirements, from delicate biomedical devices to robust aerospace structures.



Supplementary Figure S1. Comparison of inverse-designed energy curves (lines) with FEA validation (dots) for seven bistable design cases.



Supplementary Figure S2. Training loss evolution showing convergence of all loss components over 50,000 epochs for the seven inverse design cases.



Supplementary Figure S3. Evolution of normalized structural parameters during inverse design training for the seven cases.

Design targets	h_1 [mm]	h_2 [mm]	$U_{barrier}$ [mJ]	NAE (%)
$h_2 = 3.4$, $U_{barrier} = 0.05$, n = 8	0.000905	0.000691	0.049766	1.22
$h_2 = 3.9$, $U_{barrier} = 0.05$, n = 8	0.001071	0.000243	0.049728	1.06
$h_2 = 4.4$, $U_{barrier} = 0.05$, n = 8	0.000982	0.000089	0.049843	0.82
$h_2 = 3.9$, $U_{barrier} = 0.075$, n = 8	0.000950	0.000959	0.074743	0.96
$h_2 = 3.9$, $U_{barrier} = 0.1$, n = 8	0.001050	0.001725	0.099600	1.06
$h_2 = 3.9$, $U_{barrier} = 0.075$, n = 9	0.001308	0.001394	0.074578	1.39
$h_2 = 3.9$, $U_{barrier} = 0.075$, n = 10	0.002352	0.001887	0.074184	2.25

Supplementary Table S3. Target specifications and achieved values for bistable energy programming. Comparison of target and inverse-designed values for stable point heights (h_1, h_2), barrier energy ($U_{barrier}$), and normalized absolute error (NAE) for all cases in Figure 7.

Design targets	Parameter [mm]	Value
$h_2 = 3.4$, $U_{barrier} = 0.05$, n = 8	a	1.0693
	b	2.2421
	c	3.7955
	β [rad]	1.4923
$h_2 = 3.9$, $U_{barrier} = 0.05$, n = 8	a	1.3351
	b	2.4642
	c	4.2524
	β [rad]	1.5434
$h_2 = 4.4$, $U_{barrier} = 0.05$, n = 8	a	1.6380
	b	2.6089
	c	4.6543
	β [rad]	1.6016
$h_2 = 3.9$, $U_{barrier} = 0.075$, n = 8	a	1.2817
	b	2.3321
	c	4.1901
	β [rad]	1.5061
$h_2 = 3.9$, $U_{barrier} = 0.1$, n = 8	a	1.3156
	b	2.1610
	c	4.1020
	β [rad]	1.4997
$h_2 = 3.9$, $U_{barrier} = 0.075$, n = 9	a	1.0632
	b	2.1481
	c	4.2691
	β [rad]	1.4955
$h_2 = 3.9$, $U_{barrier} = 0.075$, n = 10	a	0.9292
	b	1.9656
	c	4.3098
	β [rad]	1.4991

Supplementary Table S4. Structural parameters for bistable energy programming. Inverse-designed parameters for CKO structures with specified stable points and energy barriers, including variations in h_2, barrier height, and polygon number (n = 8, 9, 10) shown in Figure 7(D) and (E).

	$\mathbf{R^2}$ (FEA)	RMSE (FEA)
$\boldsymbol{h_2 = 3.4,\ U_{barrier} = 0.05,\ n = 8}$	0.9991	0.000531
$\boldsymbol{h_2 = 3.9,\ U_{barrier} = 0.05,\ n = 8}$	0.9990	0.000562
$\boldsymbol{h_2 = 4.4,\ U_{barrier} = 0.05,\ n = 8}$	0.9993	0.000469
$\boldsymbol{h_2 = 3.9,\ U_{barrier} = 0.075,\ n = 8}$	0.9987	0.000964
$\boldsymbol{h_2 = 3.9,\ U_{barrier} = 0.1,\ n = 8}$	0.9982	0.001506
$\boldsymbol{h_2 = 3.9,\ U_{barrier} = 0.075,\ n = 9}$	0.9991	0.000808
$\boldsymbol{h_2 = 3.9,\ U_{barrier} = 0.075,\ n = 10}$	0.9992	0.000706

Supplementary Table S5. Quantitative comparison between PINN predictions and FEA simulations for all seven inverse design cases.

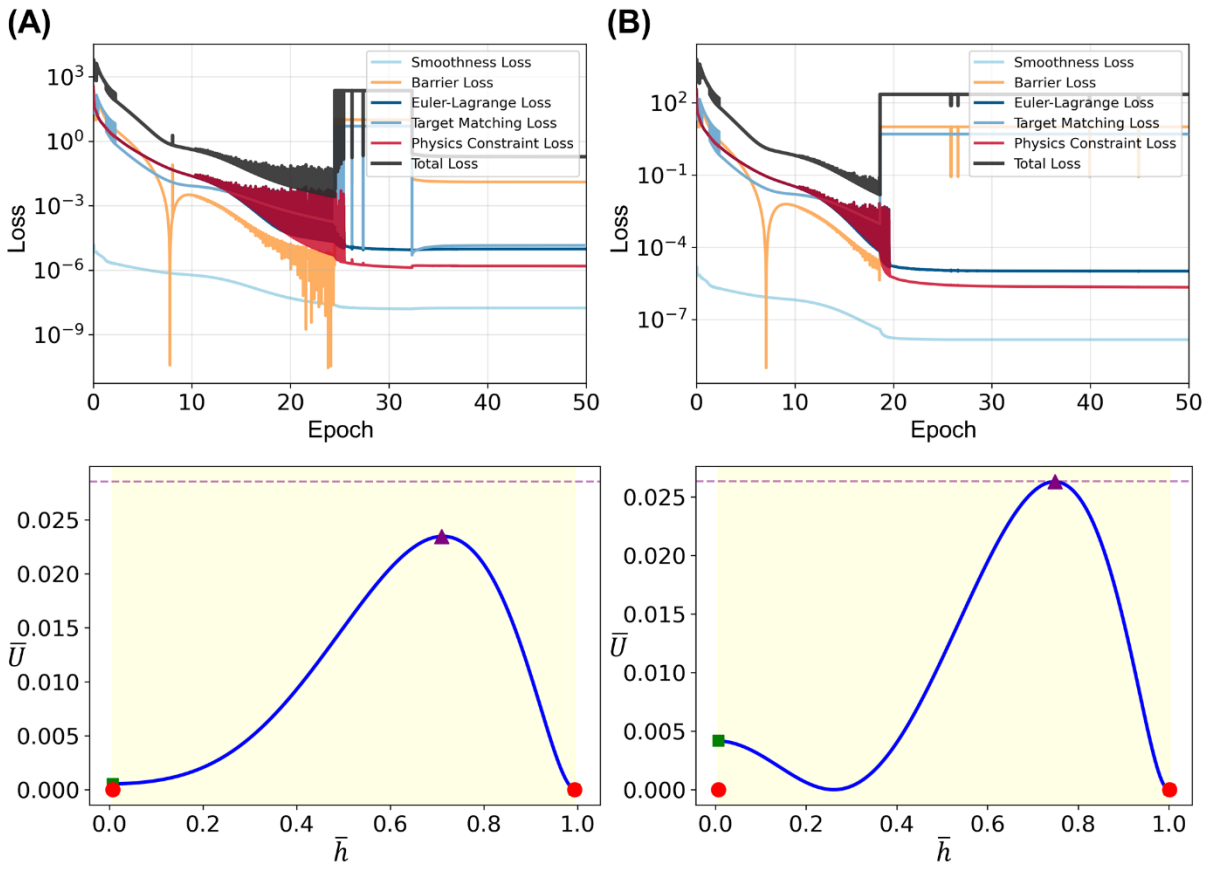
Section 5. Physical bounds on realizable energy barriers

The inverse design framework with bistable energy programming, while highly flexible, cannot achieve arbitrary combinations of stable point locations and energy barrier values due to fundamental physical constraints. This section analyzes the bounds on achievable energy barriers for given stable point configurations.

For a bistable structure with stable points at h_1 and h_2, the realizable energy barrier has both lower and upper bounds. The lower bound represents the minimum energy required for any transition between stable states, arising from inevitable crease deformation. The upper bound is more restrictive and depends on geometric constraints and kinematic limits of the structure. Supplementary Figure 4 illustrates the characteristic failure modes when target specifications exceed physical limits. Two distinct behaviors emerge when attempting to design beyond the upper bound. In the first mode (Supplementary Figure 4(A)), the optimization successfully maintains zero energy at stable points h_1 and h_2 but fails to achieve the target barrier height, resulting in a plateau below the desired value. In the second mode (Supplementary Figure 4(B)), the framework achieves the target barrier height but violates the stability constraints, producing non-zero energy at the intended stable points. The training dynamics reveal these limitations clearly. As shown in the loss evolution plots, the optimization initially converges smoothly, but upon approaching physical limits, the loss suddenly diverges. This divergence indicates that the required structural parameters would violate geometric constraints, such as c exceeding r+R or becoming less than |r-R|, or would demand kinematically impossible configurations.

These bounds have direct implications for design practice. Target specifications must be verified against physical limits before optimization. If convergence fails with behavior similar to Supplementary Figure 4(A), the stable point separation should be increased to enable higher barriers. If failure resembles Supplementary Figure 4(B), the barrier requirement is

fundamentally incompatible with the specified stable points and must be reduced. Understanding these limits prevents wasted computational effort and guides designers toward feasible specifications.



Supplementary Figure S4. Inverse design results demonstrating physical limitations. (A) Loss evolution and energy profile for a case where the target barrier height cannot be achieved while maintaining zero energy at stable points. (B) Loss evolution and energy profile for a case where zero energy at stable points cannot be maintained while achieving the target barrier height.

Section 6. Multi-layer CKO assemblies - Design parameters

This section provides the detailed design parameters for the multi-layer CKO assemblies with programmable sequential deployment presented in Section 3.4 of the main manuscript.

Design targets	Parameter [mm]	Value
Cylinder layer 1 (U=0.05)	a	3.0615
	b	3.0615
	c	6.4995
	β [rad]	1.7983
Cylinder layer 2 (U=0.08)	a	3.0615
	b	3.0615
	c	6.7476
	β [rad]	1.7433
Cylinder layer 3 (U=0.11)	a	3.0615
	b	3.0615
	c	6.9247
	β [rad]	1.7009
Cone layer 1 (U=0.05)	a	3.8268
	b	6.1229
	c	10.4319
	β [rad]	1.6587
Cone layer 2	a	2.2961

(U=0.08)	b	3.8268
	c	6.9116
	β [rad]	1.5736
Cone layer 3 (U=0.11)	a	1.5307
	b	2.2961
	c	4.5472
	β [rad]	1.5219
Zigzag layer 1 (U=0.05)	a	1.5307
	b	3.0615
	c	5.1951
	β [rad]	1.5294
Zigzag layer 2 (U=0.08)	a	1.5307
	b	3.0615
	c	5.3689
	β [rad]	1.4924
Zigzag layer 3 (U=0.11)	a	1.5307
	b	3.0615
	c	5.4883
	β [rad]	1.4626

Supplementary Table S6. Design parameters for multi-layer CKO assemblies with programmed sequential deployment. Structural parameters for three-layer configurations (cylinder, cone, zigzag) with hierarchical energy barriers of 0.05, 0.08, and 0.11 for bottom, middle, and top layers, respectively.

Section 7. Verification of Representation-Agnostic Learning via Reverse PINN

To verify that the learned energy landscape is independent of the choice of deformation coordinate, an additional analysis was conducted using a reverse-parameterized PINN, in which the rotation angle φ was treated as input and the height h was predicted as $h(\varphi)$. This test was designed to confirm that the proposed framework learns the same physical equilibrium path regardless of parameterization, thereby ensuring representation-agnostic behavior of the energy manifold.

The forward formulation learns the equilibrium trajectory $\varphi(h)$ by minimizing the total strain energy presented in the main text while satisfying the torque-free condition $\partial U/\partial\varphi = 0$. For the reverse formulation, the mapping $h(\varphi)$ was learned only within the invertible domain $|d\varphi/dh| > \varepsilon_{th}$ with $\varepsilon_{th} = 0.05$, where a one-to-one correspondence between h and φ exists. Outside this region, $d\varphi/dh \approx 0$ results in $dh/d\varphi \to \infty$, producing non-monotonic or multi-valued mappings that are physically associated with bifurcation points. Restricting the training domain to the invertible region thus guarantees that $h(\varphi)$ remains mathematically well-defined and physically meaningful.

The reverse PINN employed the same network architecture as the forward model (two hidden layers with 128 tanh-activated neurons) and was trained using a physics-constrained loss function. The loss combined the total strain energy and torque-free constraint terms, $U(h(\varphi), \varphi)$ and $\partial U/\partial\varphi = 0$, with a weak direct matching term that guided convergence toward the physically consistent branch previously identified by the forward PINN. This auxiliary term was included solely to stabilize the optimization process without altering the physical equilibrium condition or the resulting energy landscape.

$$L_{Reverse} = L_E + \lambda_{physics} L_{physics} + \lambda_m L_{match} \tag{S9}$$

$$L_E = \frac{1}{m}\sum_{i=1}^{m} U(h(\varphi_i), \varphi_i) \tag{S10}$$

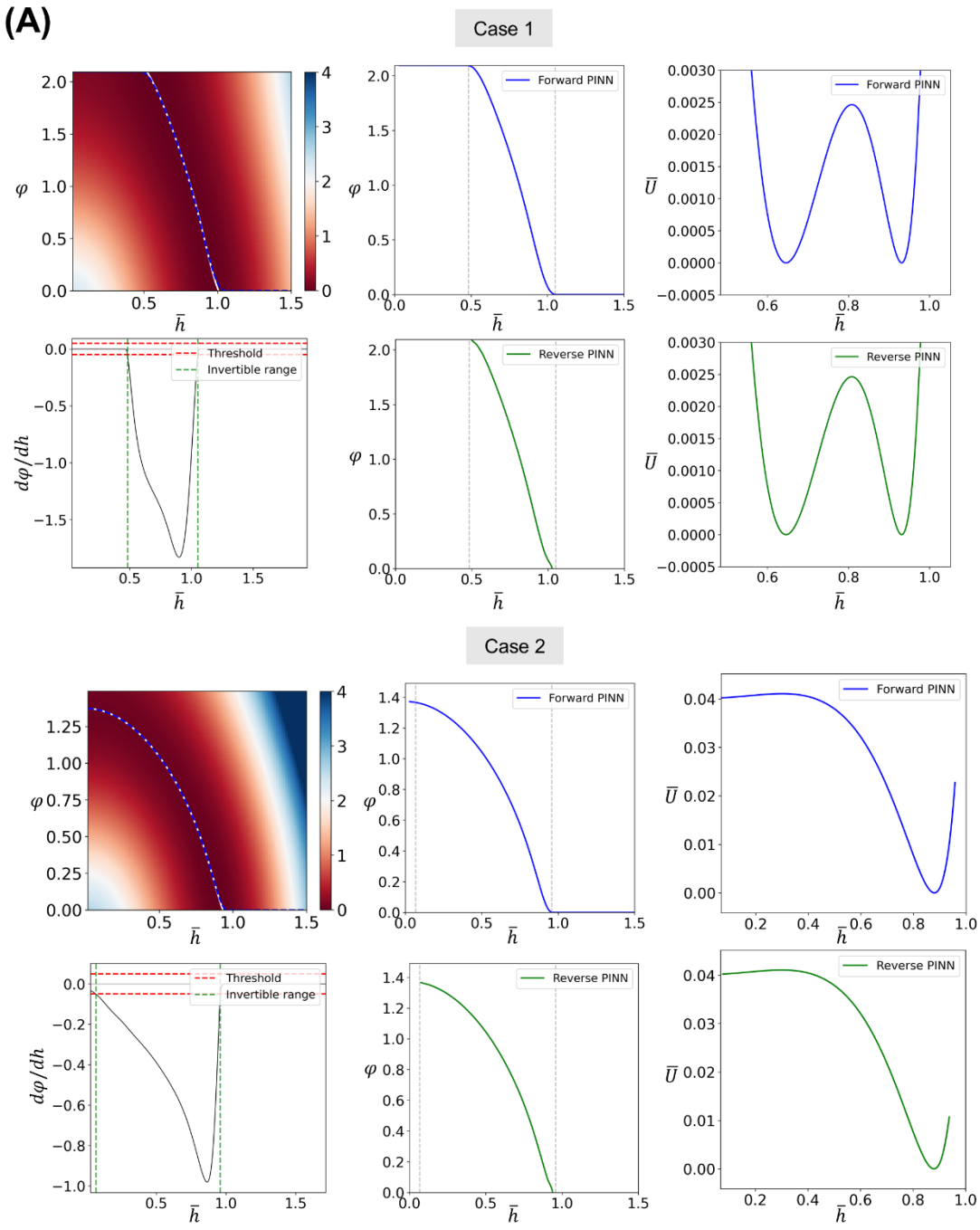
$$L_{physics} = \frac{1}{m}\sum_{i=1}^{m} \left(\frac{\partial U}{\partial \varphi}\Big|_{h(\varphi_i),\varphi_i} \right)^2 \tag{S11}$$

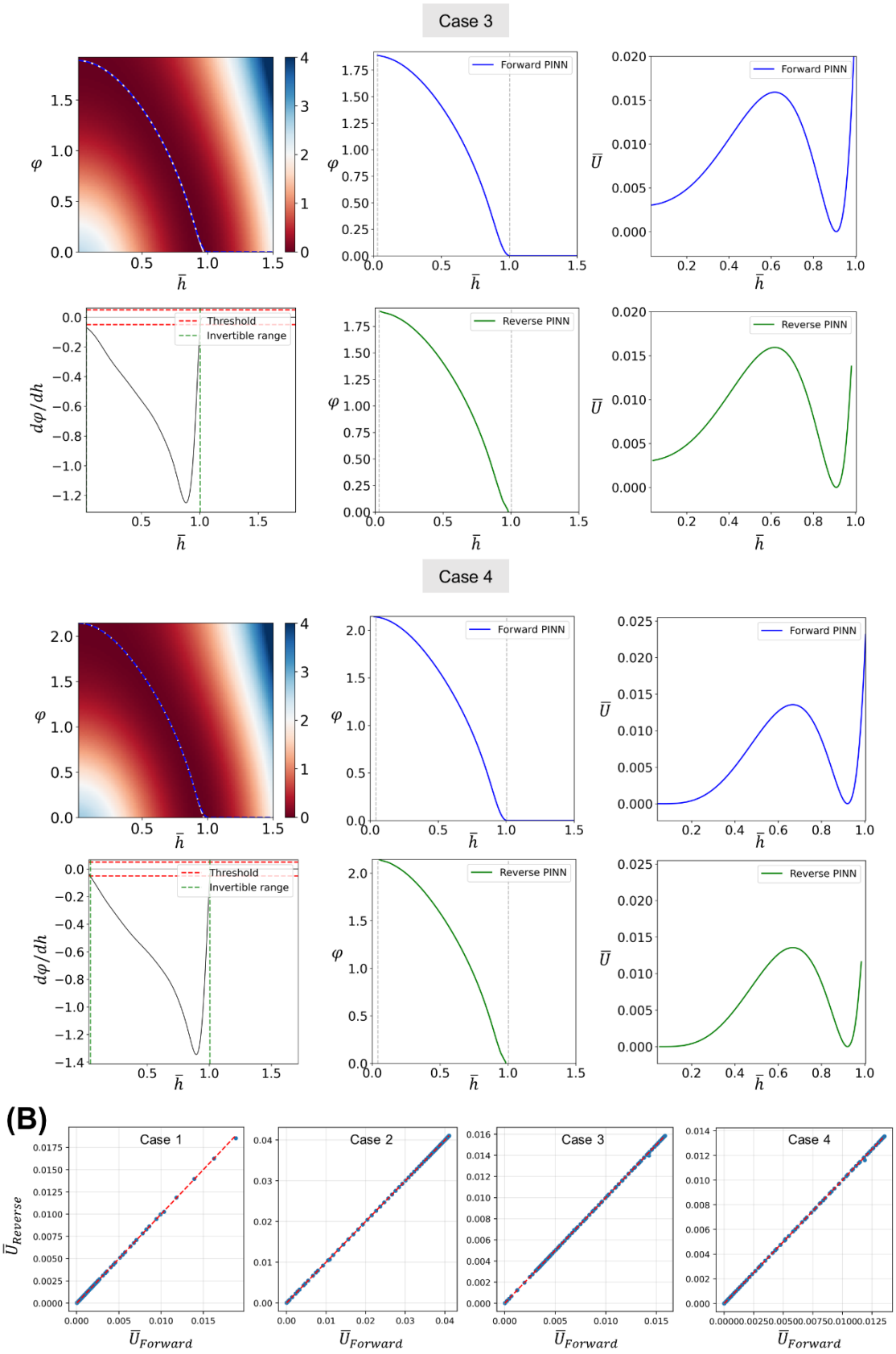
$$L_{match} = \frac{1}{m}\sum_{i=1}^{m} (h_{pred}(\varphi_i) - h_{target,i})^2 \tag{S12}$$

Supplementary Figure S5(A) compares the results of the forward and reverse formulations for four representative cases (Cases 1-4). Each case presents the two-dimensional energy field, the forward-predicted and reverse-predicted $h - \varphi$ trajectories, the normalized energy profiles, obtained from both formulations and the gradient-based invertibility map. In all cases, the forward and reverse results exhibit almost perfect overlap, confirming that both parameterizations converge to the same equilibrium path. Quantitatively, the four cases yield determination coefficients exceeding $R^2 = 0.999$ and RMSE values below 10^{-4}, demonstrating numerical equivalence between the two formulations.

Further validation is provided in Supplementary Figure S5(B), which plots the energy values obtained from the reverse formulation against those from the forward formulation for all sampling points. The data collapse precisely onto the line $y = x$, confirming one-to-one energy correspondence between the two parameterizations. This perfect alignment verifies that the total potential energy computed along the reverse trajectory is identical to that derived from the forward formulation.

These results conclusively demonstrate that the learned energy landscape is representation-agnostic within the invertible region of the deformation space. The identical energy profiles and monotonic $h - \varphi$ relations confirm that the PINN framework maintains bidirectional physical consistency under variable transformation.


(A)
Case 1
Forward PINN
Reverse PINN
Threshold
Invertible range
φ
$\bar{U}$
$d\varphi/dh$
$\bar{h}$
Case 2
Forward PINN
Reverse PINN
Threshold
Invertible range
φ
$\bar{U}$
$d\varphi/dh$
$\bar{h}$



Supplementary Figure S5. (A) Comparison between forward and reverse PINN results for four representative cases (Cases 1-4). Each case presents the two-dimensional energy field, the forward- and reverse-predicted $h-\varphi$, trajectories, the normalized energy profiles

obtained from both formulations, and the gradient-based invertibility map. (B) Comparison of the energy profiles from the forward and reverse formulations for all sampling points.

Section 8. Neural Network Architecture Justification and Training Stability

To verify the validity and stability of the neural network configuration used in this study, additional analyses were performed on the network architecture, activation function, initialization and normalization schemes, and overall convergence behavior. The purpose of this section is to provide a quantitative and physical justification for the selected architecture and training setup employed in all PINN implementations.

The network structure was first evaluated through ablation studies on the number of layers and hidden neurons using the forward PINN formulation, which allows efficient assessment of architecture-dependent convergence and computational performance. Fully connected feedforward networks with 1-4 hidden layers (each containing 128 neurons) and 2-layer networks with hidden widths of 32, 64, 128, and 256 neurons were compared. The results are summarized in Supplementary Figure S6, which presents the convergence speed and efficiency as a function of network depth and width. The final losses were nearly identical across all configurations (absolute variation $\leq 6.3 \times 10^{-4}$, approximately 0.06%), whereas convergence speed improved significantly up to two layers and 128 neurons before saturating. Specifically, the networks with 1×128, 2×128, 3×128, and 4×128 structures converged at approximately 2158 epochs (31.6 s), 2021 epochs (55.9 s), 2014 epochs (77.8 s), and 2008 epochs (95.6 s), respectively, with almost identical final losses of 0.620973, 0.620710, 0.620584, and 0.620587. Likewise, for width variation, the 2×32, 2×64, 2×128, and 2×256 models reached convergence at 2109 epochs (41.2 s), 2055 epochs (45.6 s), 2021 epochs (55.9 s), and 2015 epochs (99.3 s) with final losses between 0.620585 and 0.620655. Both studies demonstrate that performance improvement in accuracy and convergence rate saturates beyond two layers and 128 neurons, while computational cost continues to increase. Consequently, a

2-layer, 128-neuron configuration was adopted as the most efficient and stable network architecture for all forward and inverse PINN tasks.

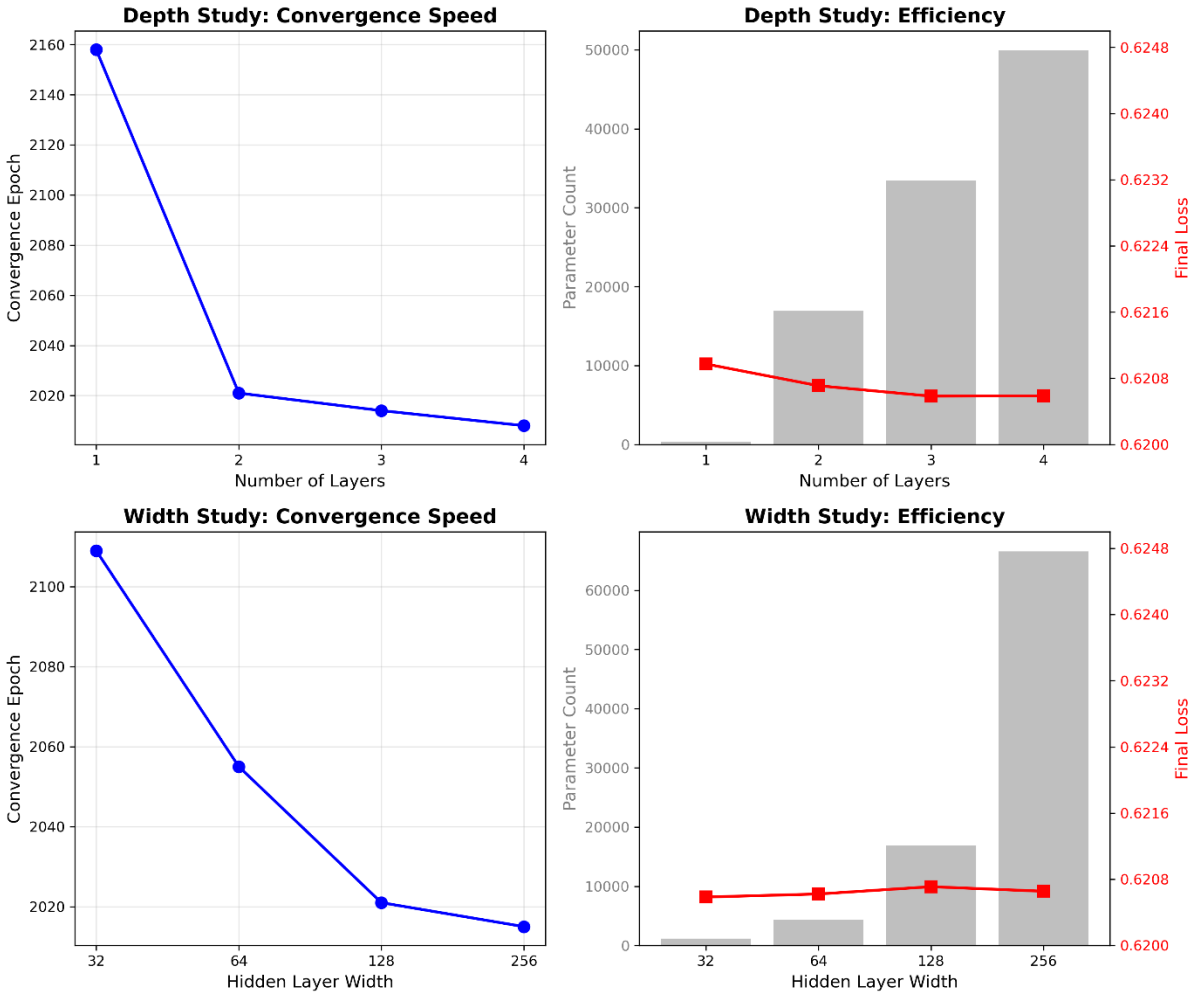
The activation function was chosen among those commonly used in physics-informed neural networks (PINNs), including ReLU, sine, GELU, and tanh. The tanh activation was selected as the most appropriate for the present formulation based on both physical and numerical considerations. The deformation mapping $\varphi(h)$ and the corresponding energy landscape $U(h, \varphi(h))$ require smooth, continuously differentiable behavior to satisfy the variational Euler-Lagrange condition, which involves second-order derivatives ($d^2\varphi/dh^2$). The tanh function is C^∞ continuous and provides smooth derivatives of all orders. Its bounded derivative ($sech^2 \in (0,1]$) suppresses gradient explosion and ensures stable learning of physics-based losses ($\partial U/\partial \varphi = 0$). Furthermore, the symmetric, S-shaped profile of tanh captures the smooth and reversible transition of the bistable Kresling structure, in contrast to ReLU, which is non-differentiable at 0 and may introduce unphysical kinks, or sine, which causes periodic oscillations. GELU adds computational complexity without observable benefit. For these reasons, tanh was adopted for all PINN models as it provides both physical consistency and numerical stability during training.

The weights were initialized using Xavier uniform initialization with zero biases to maintain balanced gradient variance across layers, ensuring stable convergence in combination with tanh activations. Input variables h were linearly scaled to their bounded domain, while the rotation output φ was restricted to the physical range using a sigmoid mapping. The total energy was normalized by $E\mathrm{A} \cdot \mathrm{c}$ to form a dimensionless energy $\overline{U} = U/EA \cdot c$, which removes unit dependence and keeps the loss terms numerically balanced. Under these settings, all training runs remained stable without any divergence or gradient explosion.

Potential failure modes were analyzed to clarify the physical limits of the model. As detailed in Supplementary Section 5, instabilities appeared only near non-invertible configurations, where multiple φ solutions exist (bifurcation-type behavior), or under unrealistically high barrier targets that exceed the mechanical stability limits of the Kresling geometry. In such cases, the optimization could not reach a torque-free equilibrium because the physical configuration itself was non-realizable. These scenarios were excluded from the training domain in the present work, ensuring that the learning process remained confined to physically meaningful and numerically stable regimes.

The convergence behavior was consistent and predictable across all tasks. In the forward PINN, convergence was typically achieved after approximately 2000 epochs, while the inverse-design formulations, including both the full energy-curve and bistable programming cases, required about 50,000 epochs for convergence. The number of epochs was not strongly affected by the geometric parameter n or by the specified barrier target. In all cases, the learning curves exhibited smooth and monotonic loss reduction, confirming stable optimization dynamics throughout training.

In summary, these analyses verify that the adopted network architecture, supported by appropriate initialization, normalization, and physically constrained energy formulation, achieves both accurate and stable training while maintaining computational efficiency.



Supplementary Figure S6. Results of the architecture ablation study in the forward PINN showing the effects of network depth and width on convergence, and final loss.