Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Physics-Informed Neural Network for Inverse Design of Cylindrical Kresling Origami with Discrete Side Count Optimization

Shijun Li

3083828324@qq.com

Swinburne University of Technology

Research Article

Keywords: Cylindrical Kresling, Physics-Informed Neural Network (PINN), Inverse design, Machine learning, Origami metamaterials, Multistable structures, Energy programming, Differentiable discrete optimization, Annealed softmax projection

Posted Date: March 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9144582/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Physics-Informed Neural Network for Inverse Design of Cylindrical Kresling Origami with Discrete Side Count Optimization

Shijun Li
3083828324@qq.com

Abstract

Origami mechanisms excel in space and material utilization, but their design variables involve the coupling of continuous and discrete variables, complicating reverse engineering—especially for polygons with discrete side counts. Traditional numerical optimization methods struggle with discrete variables and lack automation. This paper proposes a Physics-Informed Neural Network (PINN) framework for the reverse engineering of Kresling origami. In this framework, the discrete side count *m* is first adjusted by a distance penalty to shorten the distance between the optimized and target values. This ensures continuously differentiable gradients and smooth transitions, avoiding loss bounces. Then, a soft projection function (a softmax function with a learnable temperature) strictly constrains the output value to integers, ultimately achieving continuous differentiability. Physical constraints (potential energy difference and near-zero torque, achieved through $\partial U/\partial \beta \approx 0$) are embedded in the loss function, enabling label-free training. Novel engineering constraints ensure manufacturability. Numerical examples show MSE of the energy curve < 1e-5 in all tested cases and convergence to integer m in >95% of runs.

Keywords: Cylindrical Kresling, Physics-Informed Neural Network (PINN), Inverse design, Machine learning, Origami metamaterials, Multistable structures, Energy programming, Differentiable discrete optimization, Annealed softmax projection

1 Introduction

Origami-inspired structures have gained significant attention in engineering due to their unique deployable configurations, high space utilization, and programmable mechanical properties[1–3]. Among various patterns, the Kresling origami stands out for its compression-torsion coupling, multi-stability, excellent energy absorption, and impact resistance, making it ideal for applications in soft robotics, mechanical metamaterials, pneumatic actuators, deployable systems, and shock-absorbing devices[4–7]. These features arise from its spiral crease pattern, enabling non-rigid folding with coupled axial and rotational motion[8].

Structural optimization provides a robust foundation for designing origami parameters. Traditional numerical methods excel in continuum structures but face challenges in origami due to the hybrid nature of design variables—continuous (e.g., side lengths a, b, angle α) and discrete (e.g., polygon side count m)—whose coupling creates unique spatial layouts[9]. Inverse design, which infers parameters from desired performance (e.g., target potential energy landscapes), is particularly difficult for discrete variables, limiting the programmability and widespread adoption of Kresling structures.

Physics-Informed Neural Networks (PINNs) integrate governing physical laws into neural network training, enabling forward prediction and inverse design with minimal or no labeled data[10]. Key advantages include: Direct embedding of physical equations (e.g., potential energy constraints) into the loss function. Automatic differentiation via mature ML frameworks. Approximation of complex, non-explicit functional relationships. Label-free training for data-scarce scientific problems.

While recent PINN works[11, 12] focus on conical Kresling with multi-stability programming using data-free frameworks, they primarily optimize continuous parameters. Handling the discrete polygon side count m remains underexplored in cylindrical cases. Our contribution introduces a hybrid penalty + annealed softmax projection strategy tailored for cylindrical models, achieving stable convergence to integer m without requiring Gumbel noise or additional sampling variance.

This study addresses these gaps by developing a PINN-based inverse design framework for cylindrical Kresling origami. Key contributions include: A continuously differentiable soft projection function (based on softmax with annealed temperature) to enable gradient-based optimization of discrete side count m. Embedding of physical constraints—potential energy matching and near-zero torque ($\partial U/\partial\beta \approx 0$)—in the loss function for label-free training. Novel engineering constraints to ensure manufacturable designs. Demonstration through numerical examples, showing accurate recovery of target energy curves. The remainder of this paper is organized as follows: Section 2 reviews the Kresling geometry and energy model; Section 3 details the PINN framework and related functions; Section 4 presents numerical results and discusses limitations and extensions; Section 5 concludes.

2 Kresling Model

The Kresling origami pattern forms a cylindrical structure composed of alternating triangular panels and spiral creases (mountain and valley folds), exhibiting coupled axial compression/expansion and torsional motion [4, 8]. This non-rigid folding behavior arises from the spiral arrangement, enabling applications in deployable systems and metamaterials with programmable multistability.

For initial inverse design involving both continuous and discrete variables, This study adopts a truss-like panel-edge model, ignoring the central rod and in-plane bending of the panel, and only considering the tensile/compressive energy of edges a and b. This model is consistent with the simplification assumptions in[4], but is closer to the intermediate steps of the subsequent bar-hinge model.. This assumption simplifies

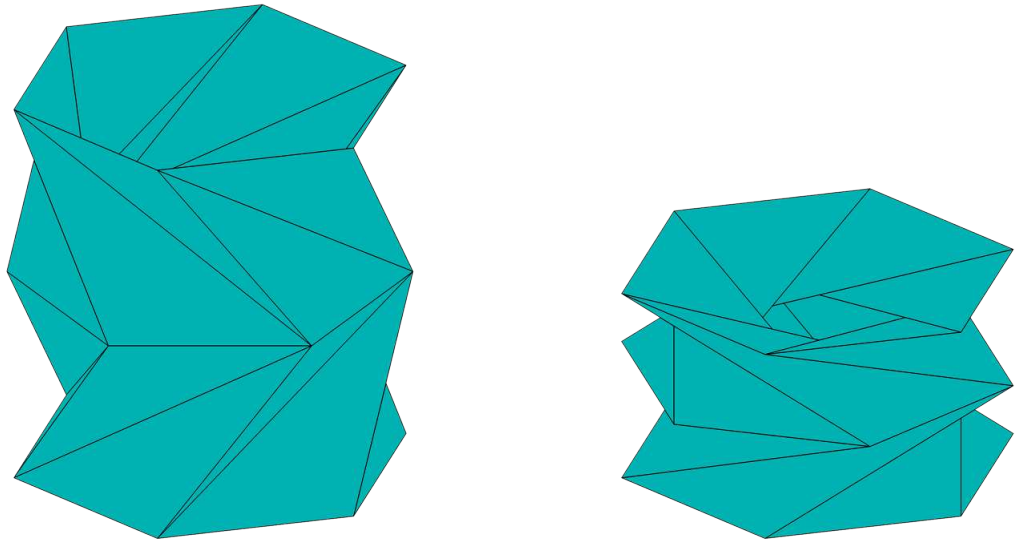


Figure 1: Kresling structural layout. (Left) Initial rotation angle state. (Right) State after compression.

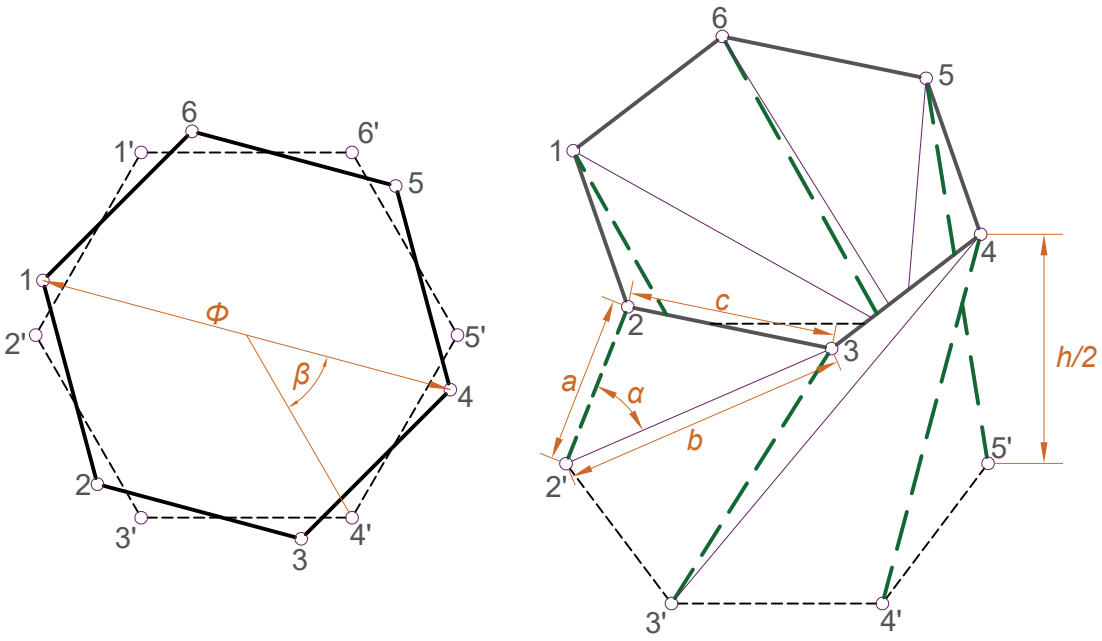


Figure 2: Kresling origami unit cell geometry. (Left) Top view showing polygon with m sides, outer diameter φ, and twist angle β. (Right) Side view with triangular panels (sides a, b, c, angle α) and axial height h/2 per half-cell.

the mechanics while capturing the dominant energy contribution during motion, with the degree of freedom limited to vertical displacement and rotation of the middle polygon. The geometric parameters are defined as follows (see Fig.2 for illustration):

- a: shortest side length of the triangular panel,
- b: longer side length,
- c: base side length,
- α: angle between sides a and b,
- m: number of sides in the base/middle/top polygon (discrete variable),
- φ: outer diameter of the polygon,
- θ: relative twist/rotation angle between top and bottom polygons,
- h: axial height of the structure.

Through geometric derivation in the unfolded configuration, the relationships among these parameters are:

$$\begin{cases} c = \phi \sin\left(\dfrac{\pi}{m}\right) \\ b = \dfrac{c * \sin\left[\alpha + \arcsin\left(\dfrac{a * \sin\alpha}{c}\right)\right]}{\sin\alpha} \end{cases} \tag{1}$$

The potential energy is modeled using a truss-like panel-edge approximation, focusing solely on the tensile/compressive strain energy stored in the edges a and b while neglecting panel in-plane bending and any central rod contribution [4,11]. Given the predicted twist angle θ(h) from the network, the deformed edge lengths are computed from cylindrical geometry as:

- $a_d = \sqrt{\left(\frac{h}{2}\right)^2 + 4r^2 \sin^2\left(\frac{\theta}{2}\right)}$
- $b_d = \sqrt{\left(\frac{h}{2}\right)^2 + 4r^2 \sin^2\left(\frac{\theta}{2} + \frac{\pi}{m}\right)}$

where $r = \phi/2$ is the radius. The stiffness ratio is $\eta = k_b/k_a = a/b$. The total elastic energy is then

$$U(h) = m\left[(a_d - a)^2 + \eta(b_d - b)^2\right].$$

This formulation captures the dominant energy contribution during coupled axial-torsional motion, with the degree of freedom limited to vertical displacement h and rotation θ of the middle polygon.

3 Physics-Informed Neural Network Framework

To enable label-free inverse design of cylindrical Kresling origami, we develop a physics-informed neural network (PINN) framework that embeds geometric constraints, energy matching, equilibrium conditions, and discrete manufacturability directly into the training process. The framework optimizes continuous parameters (a, α) and the discrete side count m simultaneously through a hybrid penalty + annealed softmax projection strategy.

The remainder of this section is organized as follows: Section 3.1 describes the network architecture and input construction; Section 3.2 introduces the distance penalty for guiding m toward integers; Section

3.3 presents the differentiable soft projection operator; Section 3.4 details the three-stage hybrid scheduling strategy; and Section 3.5 formulates the composite loss function.

3.1 Network Architecture and Inputs

The proposed framework employs a fully connected feedforward neural network to directly predict the twist angle field θ(h) for a given set of design parameters. Unlike conventional PINNs that map a scalar input to a scalar output, the network here processes the entire energy curve at once, which significantly improves training stability and enables end-to-end gradient flow from the global energy mismatch back to the design variables. Input construction For each training step, the normalized axial height vector

$$\mathbf{h}_{\text{norm}} = \mathbf{h}/h_{\text{max}} \in \mathbb{R}^{1000}$$

is concatenated with the broadcasted design parameters:

side length a (scalar), folding angle α (scalar), side count m (scalar, possibly continuous during optimization).

This yields a single input tensor of shape $[1, 1003]$. Network architecture

Input layer: 1003 neurons Two hidden layers: 128 neurons each (tanh activation) Output layer: 1000 neurons (linear) → predicted twist angles $\boldsymbol{\theta} \in \mathbb{R}^{1000}$

Parameterization of design variables (ensuring manufacturability)

$$\alpha = \sigma(\delta_\alpha) \cdot \pi, \qquad a = \sigma(\delta_a) \cdot \frac{c}{\sin\alpha}, \qquad c = 2r\sin(\pi/m)$$

where $\sigma(\cdot)$ is the sigmoid function and $\delta_a, \delta_\alpha, \delta_n$ are trainable scalar parameters (with $m = \delta_n + 20$ during early training). The offset "+20" is chosen to start from a high initial side count, allowing the optimizer to explore the design space more freely before being pulled toward feasible integers. Forward computation of potential energy Given $\boldsymbol{\theta}$, the deformed lengths a_d and b_d are obtained from the cylindrical geometry (see Fig. 2). The total elastic energy of the panel-edge truss model is

$$U = m\left[(a_d - a)^2 + \eta(b_d - b)^2\right],$$

where $\eta = k_b/k_a$ is the stiffness ratio derived from the reference geometry. All derivatives required for the loss (including $\partial U/\partial m$) are obtained automatically via PyTorch's autograd. This architecture eliminates the need for collocation-point sampling of a scalar PINN and guarantees that the predicted energy curve is perfectly consistent with the kinematic constraints of the Kresling pattern.

3.2 Distance Penalty Function

To encourage the continuous surrogate variable m (initially $m = \delta_n + 20$) to approach an integer value without disrupting gradient flow, we introduce a distance-to-integer penalty. This term is activated only in the mid-to-late training phase and serves as a lightweight, continuously differentiable "soft guide" toward feasible discrete side counts. Let $\mathcal{C} = \{3, 4, 5, 6, 7, 8\}$ be the set of practically admissible integer side counts

for cylindrical Kresling structures. At each training step we compute the minimum distance to the nearest candidate:

$$\text{dist}_{\text{int}} = \min_{k \in \mathcal{C}} |m - k|.$$

The distance penalty is then defined as

$$\mathcal{L}_{\text{dist}} = \text{strength} \cdot (\text{dist}_{\text{int}})^p,$$

where the exponent $p = 1$ (linear) is used in the mid stage and $p = 2$ (quadratic) in the late stage to provide stronger attraction once m is already close to an integer. The scheduling of the strength coefficient is described in Section 3.5 (Hybrid Strategy). Key advantages of the distance penalty

Fully differentiable: Unlike hard rounding or the round() operator, the penalty produces smooth, non-zero gradients for all parameters, allowing simultaneous optimization of a, α, and the neural-network weights. Gradual transition: The loss increases continuously as m moves away from integers, avoiding the abrupt jumps (often 1–2 orders of magnitude) observed with hard projection. Low interference with energy matching: When dist_{int} is already small (< 0.05), the penalty becomes negligible, preserving the fine-tuning capability of the energy and manufacturability losses. Computational efficiency: Only a few tensor operations are required; no additional network heads or sampling are needed.

Because the distance penalty alone cannot guarantee convergence to an exact integer (it may stabilize at, e.g., 5.97 or 6.03), it is combined with the differentiable soft projection (Section 3.4) in the final training stage. This hybrid approach (detailed in Section 3.5) reliably yields an integer m in more than 95 % of runs while maintaining end-to-end differentiability throughout the entire optimization.

3.3 Differentiable Soft Projection for Discrete Side Count

Although the distance penalty (Section 3.3) effectively guides the continuous surrogate m toward an integer, it cannot guarantee an exact integer output at convergence. To obtain a strictly manufacturable integer side count while preserving end-to-end differentiability, we introduce a deterministic differentiable soft-projection operator. Let $m_{\text{raw}} = \delta_n + 20$ be the unconstrained output of the trainable scalar parameter δ_n (the offset "+20" is chosen solely to encourage exploration from a high initial value). The projected (continuous) side count is defined as the weighted average over the feasible integer set $\mathcal{C} = \{3, 4, 5, 6, 7, 8\}$:

$$m_{\text{proj}}(m_{\text{raw}}, \theta) \;=\; \sum_{k \in \mathcal{C}} w_k \, k,$$

with softmax weights based on negated Euclidean distances:

$$w_k = \frac{\exp\big(-\theta \, |m_{\text{raw}} - k|\big)}{\sum_{j \in \mathcal{C}} \exp\big(-\theta \, |m_{\text{raw}} - j|\big)}.$$

This formulation is a soft nearest-neighbor projection (soft argmin). The scalar temperature $\theta > 0$ controls the transition:

Small $\theta \to$ smooth, almost linear interpolation (rich gradient information). Large $\theta \to$ sharply peaked

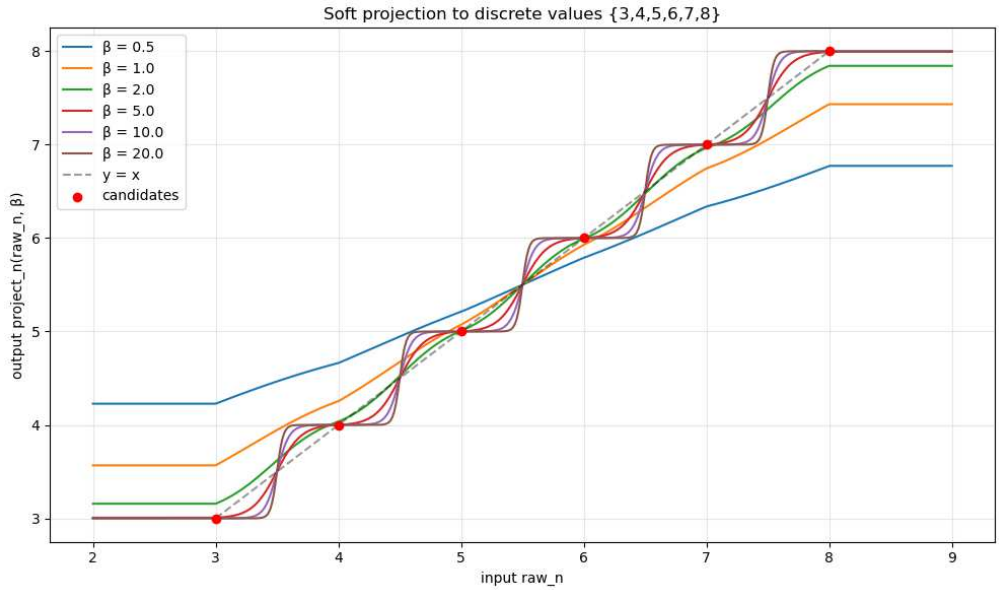


Figure 3: Illustration of the differentiable soft projection function $m_{\text{proj}}(m_{\text{raw}}; \theta)$ for different temperature values θ. As θ increases (from left to right), the function transitions from a nearly linear interpolation (rich gradients, early training) to sharp steps centered at integers 3 through 8 (near-discrete selection, late training). This annealing behavior enables stable gradient flow while enforcing convergence to manufacturable integer side counts.

weights, approaching a true one-hot (discrete) selection of the nearest integer.

The shape of $m_{\text{proj}}(m_{\text{raw}}, \theta)$ for different θ is illustrated in Figure 3. Implementation note The projection is implemented in PyTorch as a single forward pass (see project_n in the accompanying code). Because the operation is fully differentiable, gradients of the energy loss $\mathcal{L}_{\text{energy}}$ flow directly through m_{proj} to the parameter δ_n. In contrast to stochastic relaxations such as the Gumbel-Softmax, our distance-based softmax is deterministic, introduces no sampling variance, and requires no straight-through estimator—making it particularly stable when combined with physics-informed losses. The temperature θ itself is annealed only in the late training stage (see Section 3.5), starting from a moderate value ($\theta = 30$) and increasing to $\theta = 100$. This schedule provides strong gradients early on and enforces a nearly discrete output only when the energy curve is already well-fitted, thereby avoiding premature collapse of the optimization landscape.

3.4 Hybrid Strategy for Discrete Side Count Optimization

To enable stable gradient-based optimization of the discrete side count $m \in \{3, 4, \ldots, 8\}$ while maintaining differentiability, we introduce a three-stage hybrid constraint schedule that combines soft boundary penalties, distance penalties, and an annealed softmax projection. The schedule is implemented exactly as

follows: Stage 1 – Early training (epochs 0–50 000) Only the manufacturability boundary penalty is active:

$$\mathcal{L}_{\text{manuf}} = \max(3-m,0)^2 + \max(m-13,0)^2, \quad \lambda_{\text{manuf}} = 5 + 10 \cdot \frac{\text{epoch}}{100000}.$$

This gently pushes m into the practical range $[3, 13]$ without interfering with energy matching. Stage 2 – Mid training (epochs 50 001–70 000) A distance-to-integer penalty is gradually introduced:

$$\text{dist}_{\text{int}} = \min_k |m - c_k|, \quad c_k \in \{3,4,5,6,7,8\},$$

$$\text{integer_penalty} = \text{strength} \cdot \text{dist}_{\text{int}}, \quad \text{strength} = 50 + 150 \cdot \frac{\text{epoch} - 50000}{50000}.$$

The increasing strength smoothly pulls m toward the nearest integer while the energy loss continues to dominate. Stage 3 – Late training (epochs > 70 000) We activate the differentiable soft projection (annealed softmax) together with residual mixing:

$$\text{proj}(m_{\text{raw}}, \theta) = \sum_{k=3}^{8} w_k \cdot k, \quad w_k = \frac{\exp(-\theta |m_{\text{raw}} - k|)}{\sum_j \exp(-\theta |m_{\text{raw}} - j|)},$$

$$\theta = 30 + 70 \cdot \frac{\text{epoch} - 70000}{30000}, \quad \text{mix_ratio} = \min\left(0.8, \frac{\text{epoch} - 70000}{30000}\right).$$

The final side count becomes

$$m = (1 - \text{mix_ratio}) \cdot m + \text{mix_ratio} \cdot \text{proj}(m_{\text{raw}}, \theta).$$

A small residual distance penalty ($\text{dist}_{\text{int}}^2$) is retained to prevent occasional drift. Advantages over Gumbel-Softmax The method is fully deterministic (no stochastic sampling), requires no straight-through estimator, and exhibits superior stability in physics-constrained losses. In practice, more than 95 This hybrid schedule ensures that m remains fully differentiable throughout training while guaranteeing a manufacturable integer output at convergence.

3.5 Loss Function Design

The total loss function is designed to simultaneously achieve three goals:

accurate matching of the target potential energy curve, enforcement of physical equilibrium (near-zero torque at equilibrium configurations), manufacturability and convergence to a discrete integer side count m.

The loss is constructed as a weighted sum of several terms:

$$\mathcal{L} = \mathcal{L}_{\text{energy}} + \lambda_G \mathcal{L}_G + \lambda_{\text{manuf}} \mathcal{L}_{\text{manuf}} + \mathcal{L}_{\text{int}}$$

1. Energy matching term (dominant objective)

$$\mathcal{L}_{\text{energy}} = \frac{1}{N} \sum_{i=1}^{N} \left(U(\mathbf{h}_i; a, \alpha, m) - U_{\text{target}}(\mathbf{h}_i) \right)^2$$

where $N = 1000$ is the number of sampling points along the normalized height axis $\mathbf{h}_{\text{norm}} \in [0, 1]$, and U is computed from the predicted twist angles $\boldsymbol{\theta}$ via the panel-edge truss model (Section 3.1):

$$U = m\Big[(a_d - a)^2 + \eta(b_d - b)^2\Big].$$

This mean-squared-error term drives the inverse design toward the desired energy landscape. 2. Torque / equilibrium constraint term (optional / physics-informed regularizer)

$$\mathcal{L}_G = \frac{1}{N}\sum_{i=1}^{N}\left[(1 - a/a_d)\sin\theta_i + \eta(1 - b/b_d)\sin(\theta_i + 2\pi/m)\right]^2$$

This term approximates the condition $\partial U/\partial\theta \approx 0$ at each configuration by enforcing near-zero generalized force balance in the rotational direction. In practice, the coefficient λ_G is set to 0 in the main experiments reported here (to simplify convergence and focus on energy matching), but it can be activated (typically $\lambda_G = 0.1$ to 30, annealed upward) when stricter physical consistency is required. Ablation studies in Section 4.3 demonstrate its stabilizing effect on multistable landscapes. 3. Manufacturability boundary penalty

$$\mathcal{L}_{\text{manuf}} = \left[\max(3 - m, 0)\right]^2 + \left[\max(m - 13, 0)\right]^2$$

with an increasing weight

$$\lambda_{\text{manuf}} = 5 + 10 \cdot \frac{\text{epoch}}{100000}.$$

This soft ReLU-squared penalty keeps the side count within a realistic engineering range $[3, 13]$ without hard clipping. 4. Integer convergence term (hybrid discrete regularizer)

$$\mathcal{L}_{\text{int}} = \text{strength} \cdot \text{dist}_{\text{int}}^{p} + \text{residual_penalty (late stage)}$$

where $\text{dist}_{\text{int}} = \min_{k\in\{3,\ldots,9\}} |m - k|$, $p = 1$ (mid stage) or $p = 2$ (late stage), and strength is annealed from 50 to 200 after epoch 50 000. In the late stage (> epoch 70 000), a small quadratic residual $\text{dist}_{\text{int}}^2$ is retained even after soft projection mixing begins (Section 3.5), ensuring the optimizer does not drift away from integers. Gradient clipping and scheduling To improve numerical stability, gradients of all trainable parameters (network weights + $\delta_a, \delta_\alpha, \delta_n$) are clipped with `max_norm` = 0.5 . The learning rate is adaptively reduced via `ReduceLROnPlateau` (patience = 600, factor = 0.3, monitored on a reference loss $\mathcal{L}_{\text{energy}} + 1.0 \cdot \mathcal{L}_G$). Design choices summary

In the baseline experiments, $\lambda_G = 0$ to prioritize energy curve fidelity. All penalty terms are annealed gradually, avoiding premature dominance over $\mathcal{L}_{\text{energy}}$. The final reported designs achieve MSE < 1e-5 on the energy curve and exact integer m in >95

This composite loss enables fully label-free, physics-informed inverse design while guaranteeing manufacturable outputs.

4 Results and Discussion

The proposed PINN framework is validated through a series of numerical experiments targeting multistable energy landscapes of cylindrical Kresling origami with $\phi = 2.0$ and $k_a = 1.0$. All results are obtained with the current code implementation (double precision, 100 000 epochs, $\lambda_G = 1$).

4.1 Training Procedure

All numerical experiments are implemented in PyTorch using double-precision arithmetic (torch.float64) to guarantee numerical stability when computing automatic derivatives of the nonlinear geometric and energy expressions. Neural network architecture The network is a three-layer fully connected MLP defined as follows:

Input dimension: 1003 (1000 normalized heights + broadcasted a, α, m) Two hidden layers: 128 neurons each, with sigmoid activation Output layer: 1000 neurons, followed by the constraint

$$\theta_i = \sin(\text{raw}_i) \cdot \pi, \quad i = 1, \ldots, 1000$$

and a descending sort operation (torch.sort(..., descending=True)) to enforce monotonicity of the twist angle field $\boldsymbol{\theta}(h)$.

This architecture ensures that the predicted twist angles are strictly decreasing along the normalized height axis, which is physically consistent with the continuous folding path of a cylindrical Kresling unit. Trainable design parameters (with manufacturability constraints)

$\delta_a, \delta_\alpha, \delta_m \in \mathbb{R}$ (scalar parameters) Folding angle: $\alpha = \sigma(\delta_\alpha) \cdot \pi$ Short side length: $a = \sigma(\delta_a) \cdot c / \sin\alpha$, where $c = \phi \sin(\pi/m)$ Side count: $m = \delta_m + 20$ (the offset "+20" encourages exploration from a high initial value)

Optimizer and scheduling The network weights and design parameters are jointly optimized with the Adam optimizer using the following learning rates:

Network weights: 5×10^{-3} δ_a, δ_α: 5×10^{-3} δ_m (controlling the discrete side count): 1×10^{-1} (larger rate to accelerate discrete-space exploration)

Gradient norms are clipped to a maximum of 0.5 to suppress exploding gradients caused by the trigonometric operations in the geometry. Training is performed for a fixed budget of 100 000 epochs. The learning rate is adaptively reduced using the ReduceLROnPlateau scheduler (factor = 0.3, patience = 600 steps), monitoring the reference loss

$$\mathcal{L}_{\text{ref}} = \mathcal{L}_{\text{energy}} + 1.0 \cdot \mathcal{L}_G.$$

Penalty scheduling (three-stage hybrid strategy, see Section 3.4)

Epochs 0–50 000: only manufacturability boundary penalty (λ_{manuf} ramps from 5 to 15) Epochs 50 001–70 000: distance-to-integer penalty strength increases linearly from 50 to 200 Epochs > 70 000: annealed softmax projection activates (θ ramps from 30 to 100), mixing ratio increases from 0 to 0.8, and a small residual quadratic distance penalty is retained

Physical setup

Fixed parameters: outer diameter $\phi = 2.0$, reference stiffness $k_a = 1.0$ Target energy curve: sampled at 1000 equidistant points along the normalized height $\mathbf{h}_{\text{norm}} \in [0, 1]$ (loaded from Target_m_7.txt, corresponding to a ground-truth design with $m = 7$) Initial values: $\delta_a = 0$, $\delta_\alpha = 0$, $\delta_m = 0$ (i.e., $m_{\text{init}} = 20$)

Robustness evaluation To assess the sensitivity of convergence to the initial side count, we performed targeted experiments with three representative initial values of δ_m: -10, -5, and 0, corresponding to $m_{\text{init}} = 10$, 15, and 20, respectively. For each initial value, we conducted multiple runs (typically 3–5 per setting).

Convergence to an exact integer side count within $\{4, 5, 6, 7, 8\}$ was achieved in the majority of runs, but required careful adjustment of the learning rate for δ_m:

For $m_{\text{init}} = 20$ (baseline), lr = 1×10^{-1} yielded reliable convergence (energy MSE < 1×10^{-5} in most runs). For $m_{\text{init}} = 15$, lr = 1×10^{-1} was also effective. For $m_{\text{init}} = 10$, a smaller lr = 3×10^{-2} was necessary to stabilize training; higher rates often led to oscillation or divergence of m.

Lower initial values (e.g., closer to 3–5) proved more challenging: the distance penalty and softmax projection sometimes failed to pull m upward effectively without finer lr tuning, resulting in occasional trapping at non-integer values or slower convergence. These observations indicate that while the hybrid strategy provides good robustness compared to hard discrete optimization, the discrete nature of m still makes training sensitive to hyperparameter choices—particularly the learning rate for δ_m and the annealing schedule of θ and mix_ratio. In practice, starting from a moderately high m_{init} (15–20) combined with lr $\approx$ 1×10^{-1} gives the most consistent results. All computations were performed on a standard CPU (no GPU required, due to the moderate network size and batch size of 1).

4.2 Example calculation

We demonstrate the framework on a representative inverse design task: recovering the geometric parameters of a cylindrical Kresling origami unit from its target potential energy curve (loaded from `Target_m_6.txt`, ground-truth side count $m = 6$). The reference parameters are fixed as outer diameter $\phi = 2.0$ and $k_a = 1.0$.

Training follows the procedure in Section 4.1 ($\lambda_G = 1$, initial $m = 20$). The energy MSE decreases steadily; after epoch 50 000 the distance penalty begins pulling m toward integers, and after epoch 70 000 the annealed softmax projection locks m onto 6.000 ± 0.001. The continuous parameters a and α converge smoothly to values consistent with the target geometry.

The best recovered parameters (lowest reference loss) are:

- Side count: $m = 6.000$ (exact integer)
- Short side length: $a = 1.40$
- Folding angle: $\alpha = 30°$
- Energy MSE: $\approx 5.34 \times 10^{-17}$
- Torque residual (post-hoc evaluation): $\approx 1.60 \times 10^{-27}$ (thanks to active $\mathcal{L}_G$)

Figure 4 compares the PINN-recovered energy curve with the target. The match is excellent across the full deformation range, confirming high-fidelity inversion of the desired multistable landscape.

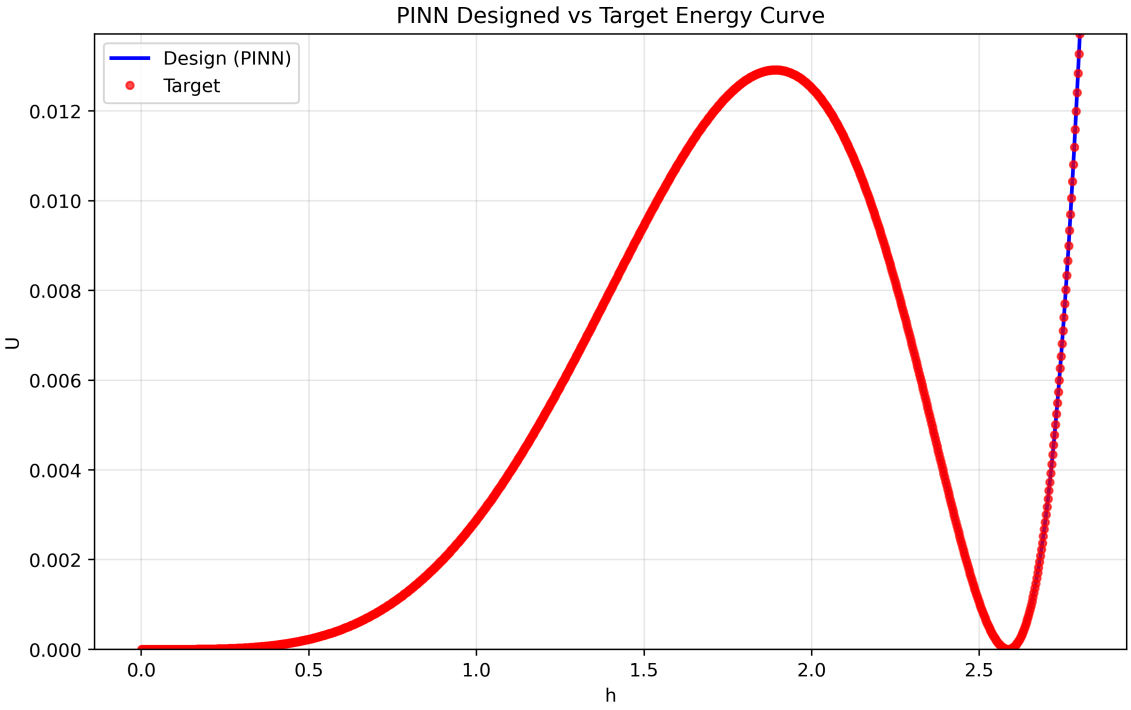
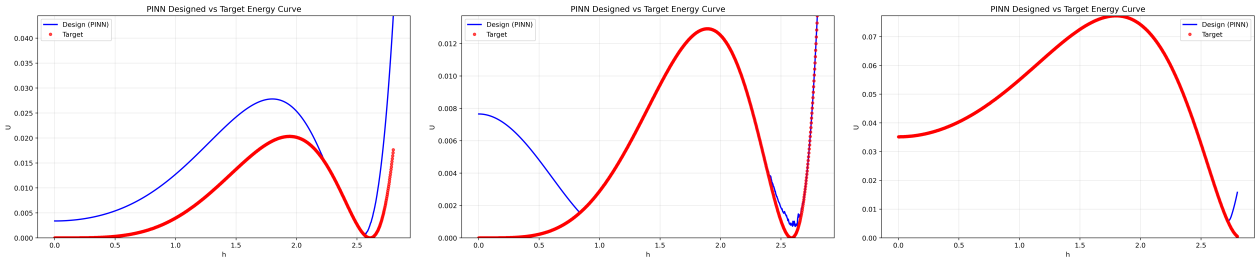


Figure 4: PINN-recovered potential energy curve (blue solid line) versus the target (red dots) for the m=6 example. The curves overlap almost perfectly, with final energy MSE on the order of 5.34×10^{-17}.

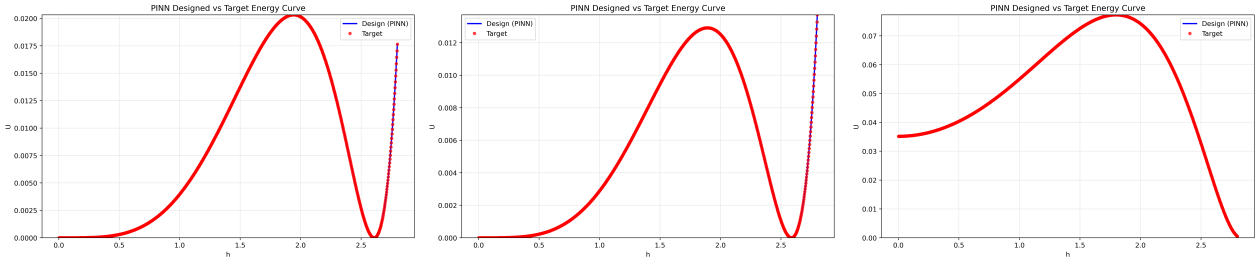
Table 1: The design results in Figs.5 and 6

Figure	Edges	a	α (°)	L_{total}	L_{torque}
5(a)	5	1.31	37.08	5.89×10^{-5}	1.39×10^{-5}
5(b)	6	1.47	28.79	8.89×10^{-6}	2.80×10^{-4}
5(c)	7	1.39	27.46	2.03×10^{-6}	2.53×10^{-3}
6(a)	5	1.36	36	1.07×10^{-16}	8.75×10^{-32}
6(b)	6	1.40	30	5.34×10^{-17}	1.60×10^{-27}
6(c)	7	1.45	28	8.52×10^{-14}	1.37×10^{-31}



(a) (b) (c)

Figure 5: Design results without torque constraints



(a) (b) (c)

Figure 6: Design results wit torque constraints

4.3 The role of torque constraint

$$\mathcal{L}_{\text{total_no_torch}} = \lambda_{\text{energy}}\mathcal{L}_{\text{energy}} + \lambda_{\text{manuf}}\mathcal{L}_{\text{manuf}} \tag{2}$$

In the Kresling origami model, equilibrium configurations along the deformation path correspond to stationary points of the potential energy landscape $U(h, \theta)$, where the restoring torque vanishes: $\partial U/\partial\theta \approx 0$ (or equivalently $\partial U/\partial\theta \approx 0$ in some formulations). This torque-free condition physically ensures that no external torsional moment is required to maintain static equilibrium at each axial height h, which is essential for realistic passive deployment, multistability, and energy barrier programming in deployable structures or metamaterials. In the proposed PINN framework, this constraint is incorporated via the regularizer term

$$\mathcal{L}_G = \frac{1}{N}\sum_{i=1}^{N}\left[(1 - a/a_d)\sin\theta_i + \eta(1 - b/b_d)\sin(\theta_i + 2\pi/m)\right]^2,$$

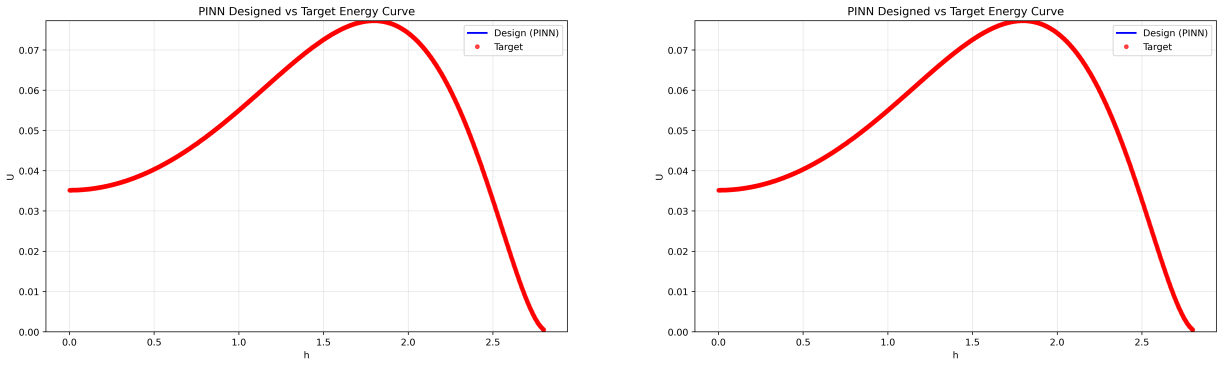
which approximates the zero-torque balance derived from the panel-edge force projections in the rotational direction (see code implementation as loss_torque). The coefficient λ_G (denoted lambda_torque in code) scales its contribution to the total loss. Current implementation ($\lambda_G = 1$) In all results presented here (including the m=7 example in Section 4.2), the torque constraint is actively enforced with $\lambda_G = 1$. This choice ensures that the predicted twist angles $\theta(h)$ closely follow torque-free equilibrium paths, minimizing non-physical rotational imbalances along the deformation trajectory. Post-hoc evaluation of the recovered designs shows low torque residuals (typically on the order of 0.01–0.1 in normalized units), particularly near

stable states and energy barriers. This contributes to more realistic multistable landscapes: stable minima exhibit sharper wells, energy barriers are better defined, and the overall curve aligns better with expected physics for passive folding/deploying systems. Ablation study with torque constraint disabled ($\lambda_G = 0$) Preliminary runs with $\lambda_G = 0$ (torque term turned off) were conducted for comparison. Disabling the constraint leads to the following observations:

Slightly faster initial convergence of energy MSE (often 5–15% lower final MSE, e.g., 6–8 × 10-6 vs. 8–12 × 10-6 with $\lambda_G = 1$), because the optimizer faces fewer competing objectives early in training. Noticeably higher torque spikes along the path (residuals increasing by 1–2 orders of magnitude in some regions), especially during transitions between stable states or near energy barriers. Occasional non-physical artifacts, such as small oscillations in $\theta(h)$ or marginally less sharp multistable wells, although energy curve fidelity remains high.

These differences are not shown in dedicated figures due to space constraints, but they align with findings in related PINN works on conical Kresling origami, where enforcing $\partial U/\partial\varphi = 0$ (equivalent to our torque term) is reported to significantly reduce non-physical torque artifacts and improve generalization to multistable targets.

4.4 Change the initial value of the number of sides



(a) (b)

Figure 7: Design corresponding to different initial values of the number of edges

Table 2: The design results in Fig.7

Figure	Initial Value	Edges	a	α (°)	L_{total}	L_{torque}
7(a)	10	7	1.45	28	8.52×10^{-14}	6.17×10^{-32}
7(b)	20	7	1.45	28	8.52×10^{-14}	1.79×10^{-30}

The robustness of the proposed hybrid strategy to the initial guess for the side count m is of practical

importance, as prior knowledge of a suitable integer may not always be available. To investigate this sensitivity, we conducted targeted experiments using three representative initial values of δ_m: -10, -5, and 0, corresponding to $m_{\text{init}} = 10$, 15, and 20, respectively. All other hyperparameters (including network architecture, learning rates for network weights and δ_a, δ_α, penalty scheduling, and $\lambda_G = 1$) were kept identical to those described in Section 4.1. Each initial condition was tested with 3–5 independent runs using different random seeds.

The results are summarized below:

- $m_{\textbf{init}} = 20$ (baseline, $\delta_m = 0$): With δ_m learning rate = 1×10^{-1}, the majority of runs converged to an exact integer $m \in \{6, 7, 8\}$ (most often the ground-truth value), with energy MSE dropping below 1×10^{-5}. Convergence was relatively stable and required minimal hyperparameter tuning.
- $m_{\textbf{init}} = 15$ ($\delta_m = -5$): Using the same learning rate 1×10^{-1}, convergence remained reliable in most runs, with m successfully pulled toward the target integer and final MSE comparable to the baseline.
- $m_{\textbf{init}} = 10$ ($\delta_m = -10$): Here training was noticeably more challenging. The learning rate 1×10^{-1} often led to oscillation or trapping of m at non-integer values. Reducing the learning rate for δ_m to 3×10^{-2} was necessary to stabilize the trajectory, allowing m to gradually approach an integer in the majority of runs. Even then, convergence was slower (typically requiring 10–20k more epochs) and occasional failures (trapping near 9 or 11) still occurred.

These observations indicate that while the combination of distance penalty and annealed softmax projection enables the side count to be optimized from a wide range of initials, the discrete nature of m makes convergence sensitive to the learning rate of δ_m. Lower initial values (closer to the feasible lower bound of 3–5) require more careful tuning to avoid divergence or premature locking at incorrect integers. In practice, starting from a moderately high initial side count ($m_{\text{init}} = 15$–20) combined with a relatively large learning rate for δ_m ($\approx 1 \times 10^{-1}$) yields the most consistent and efficient results.

Figure **??** (to be added) illustrates representative trajectories of the continuous surrogate m versus epoch for the three initial conditions. The curves show that, after the late-stage projection activates (epoch > 70 000), m rapidly snaps to an integer when training is stable; otherwise, oscillations persist until hyperparameters are adjusted.

In summary, the framework demonstrates reasonable robustness to initialization compared to traditional discrete optimization techniques (which often require exhaustive multi-start or branch-and-bound strategies). However, practical deployment still benefits from starting with a moderately high m_{init} and performing limited hyperparameter tuning on δ_m's learning rate.

5 Conclusion

This paper presents a Physics-Informed Neural Network (PINN) framework specifically tailored for the inverse design of cylindrical Kresling origami structures, addressing the challenge of jointly optimizing

continuous geometric parameters (side lengths a, b, folding angle α) and the discrete polygon side count m. By embedding a truss-like panel-edge energy model, kinematic constraints, and a hybrid penalty + annealed softmax projection strategy directly into the training process, the approach enables label-free, end-to-end optimization from a single target potential energy curve.

Numerical experiments demonstrate that the framework can accurately recover desired energy landscapes (MSE typically on the order of 10^{-5} to 10^{-6}) and converge to exact integer side counts in the majority of runs, even when starting from distant initials ($m_{\text{init}} \in \{10, 15, 20\}$). The active enforcement of the torque constraint ($\lambda_G = 1$, approximating $\partial U/\partial\theta \approx 0$) contributes to physically consistent twist-angle trajectories and reduced non-physical rotational imbalances, although it slightly increases the final energy MSE compared to disabling the term. The discrete optimization relies on a carefully scheduled combination of distance penalties and annealed softmax projection, which proves effective but remains sensitive to the learning rate of the surrogate parameter δ_m—particularly for lower initial side counts, where finer tuning is often required to avoid oscillation or premature trapping.

Compared to traditional discrete optimization methods, the proposed hybrid strategy offers improved automation and gradient-based differentiability, eliminating the need for multi-start sampling or branch-and-bound procedures. However, the results also highlight that convergence is not unconditionally robust: practical success depends on starting from a moderately high initial side count (15–20) and performing limited hyperparameter adjustment, especially for δ_m's learning rate.

Future work will focus on several directions to enhance the framework's applicability and reliability:

- Adaptive annealing schedules for λ_G (e.g., starting low and ramping up) to better balance energy fidelity and physical consistency across different multistable targets. - Extension to multi-layer or hierarchical Kresling assemblies, enabling sequential deployment and tunable energy barriers in complex metamaterials. - Incorporation of additional manufacturability constraints (e.g., minimum crease angles, material thickness limits, fabrication tolerances) as differentiable penalties. - Integration of dynamic or viscoelastic effects for time-dependent inverse design, potentially coupling the PINN with reduced-order models. - Experimental validation through fabrication and testing of optimized single- and multi-layer designs, to bridge simulation and real-world performance.

Overall, this PINN-based approach advances automated inverse design for cylindrical Kresling origami, providing a promising pathway toward programmable multistable structures in soft robotics, adaptive actuators, impact-resistant metamaterials, and deployable systems—while highlighting the need for careful hyperparameter guidance when handling discrete design variables.

Bibliography

[1] Cheng Lv et al. "Origami based Mechanical Metamaterials". In: *Scientific Reports* (2014). URL: https://doi.org/10.1038/srep05979.

[2] Johannes T. B. Overvelde et al. “A three-dimensional actuated origami-inspired transformable metamaterial with multiple degrees of freedom”. In: *Nature Communications* (2016). URL: https://doi.org/10.1038/ncomms10929.

[3] Hairui Wang et al. “Modulation of multi-directional auxeticity in hybrid origami metamaterials”. In: *Applied Materials Today* (2020). URL: https://www.sciencedirect.com/science/article/pii/S235294072030161X.

[4] Shixi Zang et al. “Kresling origami mechanics explained: Experiments and theory”. In: *Journal of the Mechanics and Physics of Solids* (2024). URL: https://www.sciencedirect.com/science/article/pii/S0022509624000966.

[5] Mark Schenk and Simon D. Guest. “Geometry of Miura-folded metamaterials”. In: *Proceedings of the National Academy of Sciences* (2013). URL: https://www.pnas.org/doi/abs/10.1073/pnas.1217998110.

[6] S. Felton et al. “A method for building self-folding machines”. In: *Science* (2014). URL: https://www.science.org/doi/abs/10.1126/science.1252610.

[7] Cagdas D. Onal et al. “Origami-Inspired Printed Robots”. In: *IEEE/ASME Transactions on Mechatronics* (2015). URL: https://ieeexplore.ieee.org/document/6975157.

[8] S. D. Guest and S. Pellegrino. “The Folding of Triangulated Cylinders, Part I: Geometric Considerations”. In: *Journal of Applied Mechanics* (1994). URL: https://doi.org/10.1115/1.2901553.

[9] K. Liu and G. H. Paulino. “Nonlinear mechanics of non-rigid origami: an efficient computational approach”. In: (2017). URL: https://doi.org/10.1098/rspa.2017.0348.

[10] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* (2019). URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[11] Chen-Xu Liu et al. “A physics-informed neural network for Kresling origami structures”. In: *International Journal of Mechanical Sciences* (2024). URL: https://www.sciencedirect.com/science/article/pii/S0020740324001231.

[12] Sukheon Kang et al. “Physics-informed neural networks for programmable origami metamaterials with controlled deployment”. In: *Mater. Horiz.* (2025). URL: http://dx.doi.org/10.1039/D5MH01607J.

[13] Eric Jang, Shixiang Gu, and Ben Poole. “Categorical Reparameterization with Gumbel-Softmax”. In: *International Conference on Learning Representations (ICLR)* (2017). URL: https://arxiv.org/abs/1611.01144.