3D Weaving with Curved Ribbons Supplementary Material

Yingying Ren[1], Julian Panetta[2], Tian Chen[1], Florin Isvoranu[1], Samuel Poincloux[1], Christopher Brandt[1], Alison Martin[2], and Mark Pauly[1]

[1]EPFL
[2]UC Davis
[3]Weaver and Independent Researcher

This supplementary document presents the detailed representations and formulas used in our weaving simulation and optimization framework and a complete set of data for each model we fabricated and validated.

1 Simulation Problem

1.1 Discrete Elastic Rod

We model each ribbon in our woven structure as a discrete elastic rod (DER) whose equilibrium shape is determined by minimizing elastic energy under the constraints imposed by its contacts with other ribbons. In the discrete elastic rods model, the elastic energy is expressed as a sum of stretching, bending, and twisting energies E_s, E_b, and E_t. We use the stretching and twisting energy proposed in [Bergou et al., 2010] and a slightly modified version of the bending energy from [Bergou et al., 2008] that was proposed in [Panetta et al., 2019]:

$$E_s = \frac{1}{2}\sum_{j=0}^{\mathtt{ne}-1} \frac{k_s^j}{\bar{l}^j}\left(|\mathbf{e}^j| - \bar{l}^j\right)^2, \tag{A1}$$

$$E_b = \frac{1}{2}\sum_{i=1}^{\mathtt{nv}-2}\sum_{j=i-1}^{i} \frac{\bar{l}^j}{2\bar{l}_i^2}\left(B_{11}({\kappa_1}_i^j - \overline{\kappa_1}_i)^2 + B_{22}({\kappa_2}_i^j - \overline{\kappa_2}_i)^2\right), \tag{A2}$$

$$E_t = \frac{1}{2}\sum_{i=1}^{\mathtt{nv}-2} \frac{\beta_i}{\bar{l}_i}(m_i - \overline{m}_i)^2. \tag{A3}$$

The supplementary material of [Panetta et al., 2019] provides a full derivation of the gradient and Hessian formulas for these energies with respect to the equilibrium and rest length variables needed for simulation. However, our curved weaving framework introduces new optimization variables that allow the ribbons' rest shapes to curve in-plane. At the same time, we constrain the rest shapes to remain planar to enable fabrication with laser cutting. We accomplish this by making $\overline{\kappa_1}_i$, the rest material curvatures in the ribbon's stiff (in-plane) direction, variables of our design optimization while setting a constant out-of-plane curvature $\overline{\kappa_2}_i = 0$.

We provide the derivative formulas for these new rest curvature variables below. Note that the rest curvature variable $\overline{\kappa_1}_i$ for vertex i appears only in the bending energy, and the terms including it involve quantities only for the incident edges $i-1$ and i. Therefore, only the following derivatives for edge index $k \in \{i-1, i\}$ are nonzero:

Partial Derivatives with respect to $\overline{\kappa_1}_i$:

$$\frac{\partial E_b}{\partial \overline{\kappa_1}_i} = -\frac{B_{11}}{2\bar{l}_i^2}\sum_{j=i-1}^{i} \bar{l}^j\left({\kappa_1}_i^j - \overline{\kappa_1}_i\right)$$

Hessian blocks involving $\overline{\kappa_{1i}}$:

$$\frac{\partial^2 E_b}{\partial \overline{\kappa_{1i}} \partial \mathbf{e}^k} = -\frac{B_{11}}{2\bar{l}_i^2} \sum_{j=i-1}^{i} \bar{l}^j \frac{\partial {\kappa_1}_i^j}{\partial \mathbf{e}^k} = -\frac{B_{11}}{2\bar{l}_i^2} \left(\bar{l}^{i-1} \frac{\partial {\kappa_1}_i^{i-1}}{\partial \mathbf{e}^k} + \bar{l}^i \frac{\partial {\kappa_1}_i^i}{\partial \mathbf{e}^k} \right)$$

$$\frac{\partial^2 E_b}{\partial \overline{\kappa_{1i}} \partial \theta^k} = -\frac{B_{11}}{2\bar{l}_i^2} \sum_{j=i-1}^{i} \bar{l}^j \frac{\partial {\kappa_1}_i^j}{\partial \theta^k} = -\frac{B_{11}}{2\bar{l}_i^2} \bar{l}^k \frac{\partial {\kappa_1}_i^k}{\partial \theta^k}$$

$$\frac{\partial^2 E_b}{\partial \overline{\kappa_{1i}} \partial \bar{l}^k} = \frac{B_{11}}{2\bar{l}_i^3} \left(\sum_{j=i-1}^{i} \bar{l}^j \left({\kappa_1}_i^j - \overline{\kappa_{1i}} \right) \right) - \frac{B_{11}}{2\bar{l}_i^2} \left({\kappa_1}_i^k - \overline{\kappa_{1i}} \right)$$

$$\frac{\partial^2 E_b}{\partial \overline{\kappa_{1i}} \partial \overline{\kappa_{1i}}} = \frac{B_{11}}{2\bar{l}_i^2} \left(\bar{l}^{i-1} + \bar{l}^i \right)$$

The equilibrium variable derivatives that appear in these expressions, $\frac{\partial {\kappa_1}_i^j}{\partial \mathbf{e}^k}$ and $\frac{\partial {\kappa_1}_i^j}{\partial \theta^k}$, are provided in the supplementary material of [Panetta et al., 2019].

1.2 Elastic Energy of Ribbon Network

1.2.1 Ribbon Network

Our woven structures consist of a collection of ribbons that interleave with each other at *crossings*. The weave's topology and initial geometric configuration are defined by a graph embedded in $\mathbb{R}^3$, where edges correspond to ribbon segments and vertices signify either crossings or free ends.

Because we assume that only two ribbons meet at a crossing, vertices of this graph must have valence at most four. We interpret valence four vertices as points where two ribbons cross each other; valence three as points where one ribbon terminates at another's interior; valence two as points where two ribbon ends meet; and valence one as a free ribbon end. In most cases, crossings involving more than two ribbons (vertex valence exceeding four) can be easily split into multiple two-ribbon crossings with a local manual modification to the input topology.

Our algorithm infers which of the ribbon segments incident each crossing belong to the same long, continuous ribbon and joins them, assigning them the local ribbon label of "A" or "B". In the valence-four case, it sorts the edges clockwise after projecting them onto the plane of best fit and picks the first and third to form ribbon A and the second and fourth to form ribbon B. In the valence-three case, it connects the segments that form the straightest possible ribbon. We additionally decide, at each crossing, whether ribbon A or B passes over the other. To promote greater contact friction and stability, we assign these using a simple breadth-first search on the input graph that ensures each ribbon alternates between passing over and under the transverse ribbons at each crossing. This algorithm is guaranteed to find a perfect alternating assignment if one exists. It turns out that any weaving pattern formed by the edges of medial polygons inscribed in a surface tessellation, for example, triangulations or quadrangulations with arbitrary singularities, is guaranteed to admit a perfect alternating assignment [Mallos, 2009]. All of the triaxial input graphs we show in the paper alternate successfully. However, our model also allows these assignments to be specified by the user for artistic purposes. Thanks to our flexible input representation, users can adjust the input graph locally rather than considering entire (potentially long and winding) ribbons.

Finally, we initialize each rod segment's rest length to the straight-line distance between its endpoints in the input graph and initialize all rest curvatures to zero.

1.2.2 Ribbon Crossings

We assume that the two ribbons at a crossing apply sufficient friction and normal forces to each other that, at equilibrium, they are in perfect contact without sliding. While this is not true for general designs, it is a desirable property for a woven structure and one that our design optimization explicitly enforces

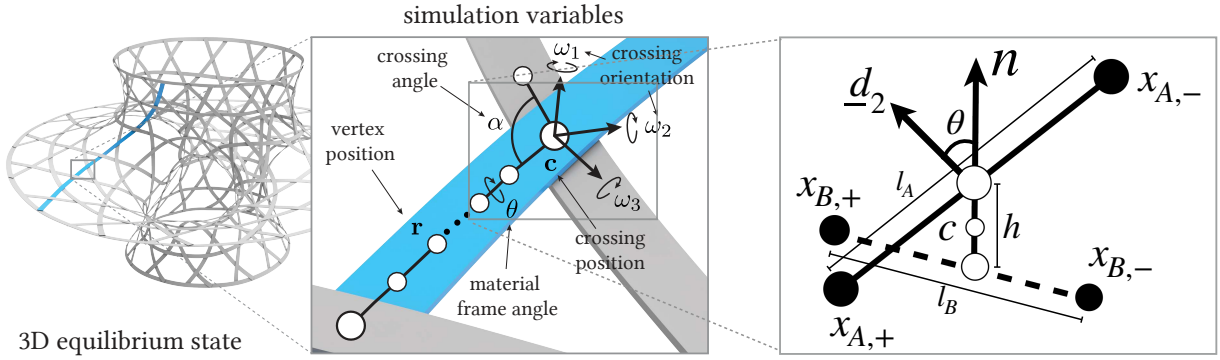

Figure 1 *Crossing Model. This is an extension of Figure 7 in the main text. The last panel shows intermediate variables needed to compute the constrained rod variables from the crossing variables.*

(Section 2.4). This allows us to model the crossings using hard constraints, forcing the two ribbons to pass through a common point $\mathbf{c}$ with common normal $\mathbf{n}$. This means we can use a crossing model very similar to the joint model proposed for simulating X-Shells in [Panetta et al., 2019]. The difference here is we found it important to account for the over/under normal offset of the ribbons at the crossings—a fundamental property of woven structures—due to their non-negligible thicknesses (see Figure 1).

A particular crossing is therefore parameterized by a position $\mathbf{c} \in \mathbb{R}^3$, an orientation represented by $\omega \in \mathbf{so}(3)$ (an angle-scaled rotation axis), the angle $\alpha \in \mathbb{R}$ between the two incident ribbons, and two lengths $l_A, l_B \in \mathbb{R}$. Length l_A (resp. l_B) specifies the length of the two overlapping discrete elastic rod edges centered at $\mathbf{c}$ that belong to each segment making up ribbon A (resp. B).

These crossing variables, combined with the over-under assignments, fully determine the configuration of all of the incident terminal discrete elastic rod edges via a nonlinear relationship. Specifically, they determine the terminal edge vectors and the ribbon normals [Panetta et al., 2019]:

$$\begin{aligned}
\mathbf{e}_A &= R(\boldsymbol{\omega})\Big(\hat{\mathbf{b}}\cos(\alpha/2) - (\hat{\mathbf{n}} \times \hat{\mathbf{b}})\sin(\alpha/2)\Big) l_A, \\
\mathbf{e}_B &= R(\boldsymbol{\omega})\Big(\hat{\mathbf{b}}\cos(\alpha/2) + (\hat{\mathbf{n}} \times \hat{\mathbf{b}})\sin(\alpha/2)\Big) l_B, \\
\mathbf{n} &= R(\boldsymbol{\omega})\hat{\mathbf{n}}.
\end{aligned}$$

The edge vectors $\mathbf{e}_A$ and $\mathbf{e}_B$ in turn determine the two centerline positions for each incident terminal edge belonging to ribbon A and B, placing them at the points

$$\begin{aligned}
\mathbf{x}_{A,\pm} &:= \mathbf{c} + s\frac{h}{2}\mathbf{n} \pm \frac{1}{2}\mathbf{e}_A \\
\mathbf{x}_{B,\pm} &:= \mathbf{c} - s\frac{h}{2}\mathbf{n} \pm \frac{1}{2}\mathbf{e}_A,
\end{aligned}$$

where h is the ribbon cross-section height (so the ribbons are held in perfect contact) and $s \in \{-1, 1\}$ indicates whether ribbon A passes under or over ribbon B. Note that the red term implementing the normal offset is new to our crossing model.

The crossing normal $\mathbf{n}$ specifies the material frame vector $\mathbf{d}_2$ for all incident terminal edges (forcing the ribbon normals to align with $\mathbf{n}$). As in [Panetta et al., 2019], this is done by setting these edges' material frame variables θ to the angle between reference frame vector $\underline{\mathbf{d}}_2$ and $\mathbf{n}$. The one difference in our implementation is that we enrich each terminal edge at each joint/crossing with an additional angle offset variable ϕ that we add in when setting θ. We set ϕ at construction time to a constant value of either $0°$ or $180°$ to handle the case of non-orientable input that the user wishes to weave with Möbius strips. Note,

however, that non-orientable inputs do not strictly require these twists: our Klein bottle example actually is *not* woven with Möbius strips, as the ribbons in question traverse the handle twice before closing up.

1.2.3 Elastic Energy

Since the crossing model does not itself store any energy, the total energy in the woven structure is the sum of each ribbon segment's elastic energy:

$$E(\mathbf{x}, \mathbf{p}) \coloneqq \sum_{r \in \mathcal{A}} E_s(\mathbf{v}_r(\mathbf{x}), \mathbf{p}) + E_b(\mathbf{v}_r(\mathbf{x}), \mathbf{p}) + E_t(\mathbf{v}_r(\mathbf{x}), \mathbf{p}), \tag{A4}$$

where $\mathcal{A}$ is the set of all ribbon segments. The vector $\mathbf{x}$ describes the deformed configuration of the entire woven structure, collecting all the crossing variables and unconstrained ribbon segment variables. The vector $\mathbf{p}$ collects all the design parameters (the rest lengths of each ribbon segment and the rest curvatures of each vertex). The nonlinear function $\mathbf{v}_r$ computes the centerline positions and material frame angles that describe the deformed configuration of the DER representing ribbon segment r: for "interior" DER edges/vertices this function simply extracts the corresponding variables from $\mathbf{x}$ while for "terminal" edges/vertices this function implements the formulas discussed in Section 1.2.2.

This energy has the same structure as the X-Shell energy in [Panetta et al., 2019], apart from our enrichments of $\mathbf{p}$ and $\mathbf{v}_r$. Therefore the chain rule formulas they present for the gradients and Hessians in Section 1.5 of their supplement apply to our energy as well. We simply use our formulas from Section 1.1 for the new gradient and Hessian blocks corresponding to the rest curvature variables in $\mathbf{p}$ and add additional terms to the Jacobian $\frac{\partial v_k}{\partial x_i}$ and Hessian $\frac{\partial^2 v_k}{\partial x_i \partial x_j}$ that originate from the ribbons' normal offset, $\pm s\frac{h}{2}\mathbf{n}$. These additional terms follow directly from the expressions given for $\frac{\partial \mathbf{n}}{\partial \boldsymbol{\omega}}$ and $\frac{\partial^2 \mathbf{n}}{\partial \boldsymbol{\omega} \partial \boldsymbol{\omega}}$ in Section 2 of [Panetta et al., 2019]'s supplement.

1.2.4 Rigid Motion and Alignment

We pin down rigid motion and ensure that our simulated structure remains optimally aligned with the target surface during the design optimization by adding a small force attracting points in the structure to their closest points on the target surface. We achieve this by adding a surface fitting term to the elastic energy to define a total potential energy function:

$$U(\mathbf{x}, \mathbf{p}) \coloneqq E(\mathbf{x}, \mathbf{p}) + \frac{\beta_U}{l_0^2} T(\mathbf{x}),$$

where T is the surface fitting function defined below in (A6).

Another option would be to pin down rigid motion by fixing the position and orientation of one of the crossings as in [Panetta et al., 2019]. This is undesirable for two reasons: first, the chosen crossing may be positioned suboptimally with respect to the target surface, and this cannot be corrected by the design optimization if it is held fixed. Second, small twisting and bending deformations in the neighborhood of this fixed crossing (caused, e.g., by tiny design variables changes) are amplified into large motions at the opposite end of the structure, causing robustness issues for minimizing our surface fitting objective. We could address both of these issues by incorporating in this fitting objective a registration of the target surface to the current equilibrium (for example, using ICP [Pottmann et al., 2006]), but this would add significant complexity to the problem.

For the final simulation, after the design optimization completes, we can still ensure our structure reproduces the target shape without these small fictitious forces by setting $\beta_U = 0$ and minimizing only the elastic energy (fixing a single crossing).

2 Design Optimization

We formulate the curved weaving design problem as minimizing the following objective function:

$$J(\mathbf{p}) = \frac{\beta_T}{l_0^2} T(\mathbf{x}_\mathbf{p}^*) + \frac{1}{U_0} U(\mathbf{x}_\mathbf{p}^*, \mathbf{p}) + R(\mathbf{p}) + \beta_c \mathcal{C}(\mathbf{x}_\mathbf{p}^*, \mathbf{p}). \tag{A5}$$

Here R is a regularization term promoting smooth designs, and $\mathcal{C}$ penalizes designs with problematic ribbon forces at the crossings (Section 2.4).

The vector $\mathbf{x}_\mathbf{p}^*$ contains the equilibrium values of the structure's deformation variables with the given design parameter vector $\mathbf{p}$. In other words, $\mathbf{x}_\mathbf{p}^*$ minimizes the total potential energy function U. We apply constraints on $\mathbf{p}$ such that the rest curvature variables take value between -2π and 2π and that the rest length variables are above a small positive value ϵ. The parameter β trades off between finding a low-energy design and fitting the structure to the desired shape. It is set much larger than the weights for the other terms since this is the main objective of the outer loop. The normalization constant l_0 is set to the target surface's bounding box diagonal, and constants E_0 and R_0 are the initial values of the corresponding terms.

We decrease the parameter β_U in U as the optimization progresses to ensure that the final structure is in equilibrium without any fictitious forces.

We can substitute the gradient and Hessian of our model's elastic energy into the design objective derivative expressions defined in the supplementary of [Panetta et al., 2019] Section 3 and add the gradient and Hessian of the regularization terms to obtain the derivative information needed here.

2.1 Surface Fitting

The surface fitting function is defined similar to [Panetta et al., 2019] as follows:

$$T(\mathbf{x}) := \frac{1}{2}\|\mathbf{x} - P_{\text{surf}}(\mathbf{x})\|_{W_{\text{surf}}}^2 + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_{\text{tgt}}\|_W^2, \tag{A6}$$

where $P_{\text{surf}}(\mathbf{x})$ projects the centerline positions of the ribbon segments in the woven structure onto the target surface. For X-shells, only the projection of the joint positions was used. Here, we expose the choice between using crossing positions and centerline positions as a parameter in the implementation.

2.2 Optimization Stage I

As explained in Section 7 of the main text, we propose a simplified formulation of the design optimization that rapidly converges to a sensible initial design approximating the target surface without needing to solve for the equilibria of badly-behaved designs. However, since it does not compute exact equilibria, it is unable to tightly fit the input surface and optimize contact forces like our full optimization.

This formulation minimizes a combination of the structure's elastic energy and our regularization terms over both the equilibrium and design variables while constraining the weave's crossings to lie at their original positions from the embedded input graph (which we assume already lies on the target surface).

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{p}} E(\mathbf{x}, \mathbf{p}) + R(\mathbf{p}) \\ &\text{s.t. } S_\mathcal{C}\mathbf{x} = \jmath_{tgt} \\ &\quad S_{rl}\mathbf{p} > \epsilon \end{aligned} \tag{A7}$$

where $S_\mathcal{C}$ and S_{rl} are matrices selecting the crossing position variables and the rest length parameters. The parameters μ and η are the weights for the two regularization terms. We normalize the elastic energy term by the initial value of the elastic energy stored. The function R is a regularization term defined below.

2.3 Regularization

We define two regularization components: $R(\mathbf{p}) = \beta_\kappa \mathcal{S}(\mathbf{p}) + \frac{\mu}{\mathcal{F}_0}\mathcal{F}(\mathbf{p})$. The parameter β_l in equation (5) of the main text is equal to $\frac{\mu}{\mathcal{F}_0}$, where normalization factor $\mathcal{F}_0$ is the total initial rest length of the weave, and μ is a user-specified weight.

2.3.1 Regularization Terms' Gradient and Hessian

Smoothing

$$\mathcal{S}(\mathbf{p}) = \frac{1}{2}\sum_{j=0}^{\texttt{ne}-1}(\overline{\kappa_1}_j - \overline{\kappa_1}_{j+1})^2, \tag{A8}$$

where the sum ranges over all `ne` discrete elastic rod edges in the weave.

$$\frac{\partial \mathcal{S}}{\partial \overline{\kappa_1}_i} = \begin{cases} \overline{\kappa_1}_i - \overline{\kappa_1}_{i-1}, & i = 1 \\ \overline{\kappa_1}_i - \overline{\kappa_1}_{i+1}, & i = nv - 2 \\ 2\overline{\kappa_1}_i - \overline{\kappa_1}_{i-1} - \overline{\kappa_1}_{i+1}, & \text{otherwise} \end{cases}$$

$$\frac{\partial^2 \mathcal{S}}{\partial \overline{\kappa_1}_i^{\,2}} = \begin{cases} 1, & i = 1, nv - 2 \\ 2, & \text{otherwise} \end{cases}$$

$$\frac{\partial^2 \mathcal{S}}{\partial \overline{\kappa_1}_i \partial \overline{\kappa_1}_{i\pm1}} = -1$$

Note that the standard discrete elastic rod model does not assign curvature quantities to (or accumulate beinding energies from) the first and last vertex. This would mean the first and last edges of each ribbon segment do not contribute to our smoothing regularization. However, when we assemble the ribbon segments into a network, the segments connecting at a crossing are meant to function as a larger unified ribbon. Hence we *do* need to accumulate energy for these connecting terminal edges. We achieve this by copying the value of, e.g., the second-to-last curvature variable of one ribbon segment to the first curvature variable of the segment it connects with.

Rest Length Regularization

$$\mathcal{F}(\mathbf{p}) = \|S_{\texttt{rl}}\mathbf{p}\|_1, \tag{A9}$$

where the matrix $S_{\texttt{rl}}$ selects all rest length variables from the design parameter vector $\mathbf{p}$,

$$\frac{\partial \mathcal{F}}{\partial \bar{l}^j} = 1.$$

2.4 Contact Optimization

Our crossing model (Section 1.2.2) is only appropriate for woven structures whose crossings feature compressive forces and sufficient friction to overcome any tangential elastic forces; otherwise some ribbons of the physical model must be held together with pins to reproduce the simulated equilibrium. While adding some pins may be practical, adding too many is tedious and can undermine the purity of the weaving aesthetic.

The elastic energy term in the design optimization objective already prefers to reduce sliding and separation forces at the crossings; these forces act to deform the ribbons, increasing their energy, and are thus penalized by the objective. However, this effect is indirect and imperfect (beneficial compressive forces are likewise penalized), and several of our examples still feature problematic forces at the crossings after stages I and II. We formulate additional terms that explicitly promote well-behaved crossings.

Consider the j^{th} crossing whose position is given by deformation variable $\mathbf{c}_j$ and whose incident rod segments composing the ribbons locally labeled A and B are collected in the sets $\mathcal{A}_j^A$ and $\mathcal{A}_j^B$, respectively. First, we measure the force $\mathbf{f}_j$ of ribbon A acting on ribbon B by summing the elastic energy's gradient with respect to crossing position $\mathbf{c}_j$ *over only the incident ribbon segments belonging to* B:

$$\mathbf{f}_j = \frac{\partial}{\partial \mathbf{c}_j} \underbrace{\sum_{r \in \mathcal{A}_j^B} E_s(\mathbf{v}_r(\mathbf{x}), \mathbf{p}) + E_b(\mathbf{v}_r(\mathbf{x}), \mathbf{p}) + E_t(\mathbf{v}_r(\mathbf{x}), \mathbf{p})}_{\tilde{E}_j}.$$

Next, we can compute the normal *separation* force as:

$$f_j^n := s_j \mathbf{n}_j \cdot \mathbf{f}_j,$$

where s_j is the crossing's over/under sign introduced in Section 1.2.2. Positive values of this scalar force measure correspond to forces pulling the ribbons apart. The squared tangential force magnitude can be computed similarly:

$$\left(f_j^t\right)^2 := \|\mathbf{f}_j\|^2 - \left(f_j^n\right)^2.$$

Finally, our contact optimization term is formulated as:

$$\mathcal{C}(\mathbf{x}, \mathbf{p}) := \frac{1}{2} \sum_j w_n \left(f_j^n - \epsilon_n\right)_+^2 + w_t \left(f_j^t\right)^2,$$

where $(\cdot)_+$ clamps its argument to positive values, ϵ_n is an "activation threshold" for the separation force penalty, and w_n and w_t are weights for the normal and tangential terms. Our implementation actually supports setting different normal/tangential weights for the internal (valence four) and the boundary (valence 2 or 3) crossings, whose stability can be more important. Note that setting ϵ slightly *negative* will promote a strictly compressive force rather than just eliminating tensile forces.

This objective term differs from the rest in that it depends on the *gradient* of individual rods' elastic energies at the structure's equilibrium. This means its gradient involves the elastic energy Hessian, and its second derivatives (needed for the design objective Hessian-vector product) will involve third derivatives of the elastic energy. However, it turns out, like for the rest of the design optimization, we need only *directional* derivatives of the elastic energy Hessian that can be computed efficiently with automatic differentiation.

We show here how to compute $\frac{\partial f_j^n}{\partial \mathbf{x}}$ and $\frac{\partial f_j^n}{\partial \mathbf{p}}$; from these, $\frac{\partial \mathcal{C}}{\partial \mathbf{x}}$ and $\frac{\partial \mathcal{C}}{\partial \mathbf{p}}$ can be computed simply by the chain rule. Directly differentiating f_j^n, we find:

$$\frac{\partial f_j^n}{\partial \mathbf{x}} = s_j \left(\mathbf{f}_j \cdot \frac{\partial \mathbf{n}_j}{\partial \mathbf{x}} + \mathbf{n}_j \cdot \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}}\right) = s_j \left(\mathbf{f}_j \cdot \frac{\partial \mathbf{n}_j}{\partial \mathbf{x}} + \mathbf{n}_j \cdot \frac{\partial^2 \tilde{E}}{\partial \mathbf{c}_j \partial \mathbf{x}}\right),$$
$$\frac{\partial f_j^n}{\partial \mathbf{p}} = s_j \mathbf{n}_j \cdot \frac{\partial \mathbf{f}_j}{\partial \mathbf{p}} = s_j \mathbf{n}_j \cdot \frac{\partial^2 \tilde{E}}{\partial \mathbf{c}_j \partial \mathbf{p}}.$$

In these formulas, $\frac{\partial \mathbf{n}_j}{\partial \mathbf{x}}$ has a single nonzero 3×3 block corresponding to crossing j's orientation variable, containing $\frac{\partial \mathbf{n}_j}{\partial \omega_j}$; a formula for this is given in [Panetta et al., 2019]'s supplement. The other terms use blocks of the Hessian formulas for the discrete elastic rods energy.

Finally, the contribution to the full gradient can be computed,

$$\frac{\mathrm{d}\mathcal{C}(\mathbf{x}_\mathbf{p}^*, \mathbf{p})}{\mathrm{d}\mathbf{p}} = \frac{\partial \mathcal{C}}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}_\mathbf{p}^*}{\mathrm{d}\mathbf{p}} + \frac{\partial \mathcal{C}}{\partial \mathbf{p}} = -\mathbf{y}^T \frac{\partial^2 U}{\partial \mathbf{x} \partial \mathbf{p}} + \frac{\partial \mathcal{C}}{\partial \mathbf{p}},$$

where $\mathbf{y}$ is the adjoint state solving

$$\frac{\partial^2 U}{\partial \mathbf{x}^2} \mathbf{y} = \frac{\partial \mathcal{C}}{\partial \mathbf{x}}.$$

The Hessian-vector product for $\mathcal{C}$ can then be computed for a design perturbation $\mathbf{p}$:

$$\frac{\partial^2 \mathcal{C}(\mathbf{x}_\mathbf{p}^*, \mathbf{p})}{\partial \mathbf{p}^2}\delta\mathbf{p} = -\delta\mathbf{y}^T \frac{\partial^2 U}{\partial \mathbf{x}\partial \mathbf{p}} - \mathbf{y}^T \left(\frac{\partial^3 U}{\partial \mathbf{x}\partial \mathbf{p}\partial \mathbf{x}}\delta\mathbf{x} + \frac{\partial^3 U}{\partial \mathbf{x}\partial \mathbf{p}\partial \mathbf{p}}\delta\mathbf{p} \right) + \frac{\partial^2 \mathcal{C}}{\partial \mathbf{p}\partial \mathbf{x}}\delta\mathbf{x} + \frac{\partial^2 \mathcal{C}}{\partial \mathbf{p}^2}\delta\mathbf{p},$$

where $\delta\mathbf{y}$ is the solution to:

$$\frac{\partial^2 U}{\partial \mathbf{x}^2}\delta\mathbf{y} = \frac{\partial^2 \mathcal{C}}{\partial \mathbf{x}^2}\delta\mathbf{x} + \frac{\partial^2 \mathcal{C}}{\partial \mathbf{x}\partial \mathbf{p}}\delta\mathbf{p} - \left(\frac{\partial^3 U}{\partial \mathbf{x}\partial \mathbf{x}\partial \mathbf{x}}\delta\mathbf{x} + \frac{\partial^3 U}{\partial \mathbf{x}\partial \mathbf{x}\partial \mathbf{p}}\delta\mathbf{p} \right) \mathbf{y}.$$

We compute the directional derivatives $\frac{\partial^2 \mathcal{C}}{\partial \mathbf{x}^2}\delta\mathbf{x} + \frac{\partial^2 \mathcal{C}}{\partial \mathbf{x}\partial \mathbf{p}}\delta\mathbf{p}$ and $\frac{\partial^2 \mathcal{C}}{\partial \mathbf{p}\partial \mathbf{x}}\delta\mathbf{x} + \frac{\partial^2 \mathcal{C}}{\partial \mathbf{p}^2}\delta\mathbf{p}$ with a (scalar) automatic differentiation of our implementation of $\frac{\partial \mathcal{C}}{\partial \mathbf{x}}$ and $\frac{\partial \mathcal{C}}{\partial \mathbf{p}}$.

3 Numerical Solver

The Equilibrium problem and the Stage I optimization problem are solved using a customized Newton-based solver used in [Panetta et al., 2019]. One of the main difficulties of the simulation problem and Stage I is handling saddle points in the energy landscape where the Hessian matrix becomes indefinite. In such cases, we modify the Hessian by using a physically meaningful mass matrix. Please see [Panetta et al., 2019] Section 4 for more details.

We solve the Stage II and Stage III optimization problems using a Newton-CG trust region method. This method converges faster compared to quasi-Newton methods and we use the implementation from [Artelys, 2019] for efficiency.

4 Experiment Settings

The following are the weight settings that we found to work empirically on the results shown in the paper. We are able to use a simple set of parameters for all the results shown thanks to the normalization parameters we defined above and in the main text.

All stages The rest length lower bound ϵ is set to be 0.125 multiplied by the minimum segment rest length from the input graph.

Stage I The smoothing weight β_κ is set to be 0.1, and the rest length sum weight μ is set to be 0.01.

Stage II The smoothing weight β_κ is set to be 10, and the rest length sum weight μ is set to be 0.1. As explained in the main text, we run multiple phases of this optimization with different attraction weight β_U in the potential energy term. For all models shown in the main text, the sequence of attraction weights used is $100, 10^{-3/2}$, and 10^{-5}. The weight for the target-surface-fitting term β_T is set to 5×10^5 by default. At the begining of each phase, if the potential energy or regularization terms exceed the target-surface-fitting term, and if the maximum distance from the weave to the target surface is larger than 5% of the bounding box diagonal, then we raise β_T to bring the target-surface-fitting term up to the largest term.

Stage III The smoothing weight β_κ, rest length sum weight μ, and the attraction weight β_U remain unchanged after stage II. The weight for the contact term β_c is set to 5×10^6 by default. The normal weight w_n is set to 10, and the tangential weight w_t is set to 1. We compute the (signed) normal separation forces at each crossing after stage II and set ϵ_n to be their 75^{th} percentile (clamped to be ≤ 0). All forces at the boundary crossings are ignored, i.e., their weights are zero. Like with β_U in stage II, if the objective value of the contact term is less than any other term, we increase β_c to bring it equal to the highest value.

5 Data

In the following figures, we illustrate the inputs, simulation of the optimized woven model, the laser cutting file for fabricating the ribbons and the physical models for eleven examples. In the color bar for visualizing the distance from the scan output to the target surface, the parameter w is the average ribbon width.

References

[Artelys, 2019] Artelys (2019). Artelys knitro - nonlinear optimization solver.

[Bergou et al., 2010] Bergou, M., Audoly, B., Vouga, E., Wardetzky, M., and Grinspun, E. (2010). Discrete viscous threads. *ACM Transactions on Graphics (TOG)*, 29(4):1–10.

[Bergou et al., 2008] Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., and Grinspun, E. (2008). Discrete elastic rods. In *ACM transactions on graphics (TOG)*, volume 27, page 63. ACM.

[Mallos, 2009] Mallos, J. (2009). How to weave a basket of arbitrary shape.

[Panetta et al., 2019] Panetta, J., Konaković-Luković, M., Isvoranu, F., Bouleau, E., and Pauly, M. (2019). X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)*, 38(4):1–15.

[Pottmann et al., 2006] Pottmann, H., Huang, Q.-X., Yang, Y.-L., and Hu, S.-M. (2006). Geometry and convergence analysis of algorithms for registration of 3d shapes. *International Journal of Computer Vision*, 67(3):277–296.

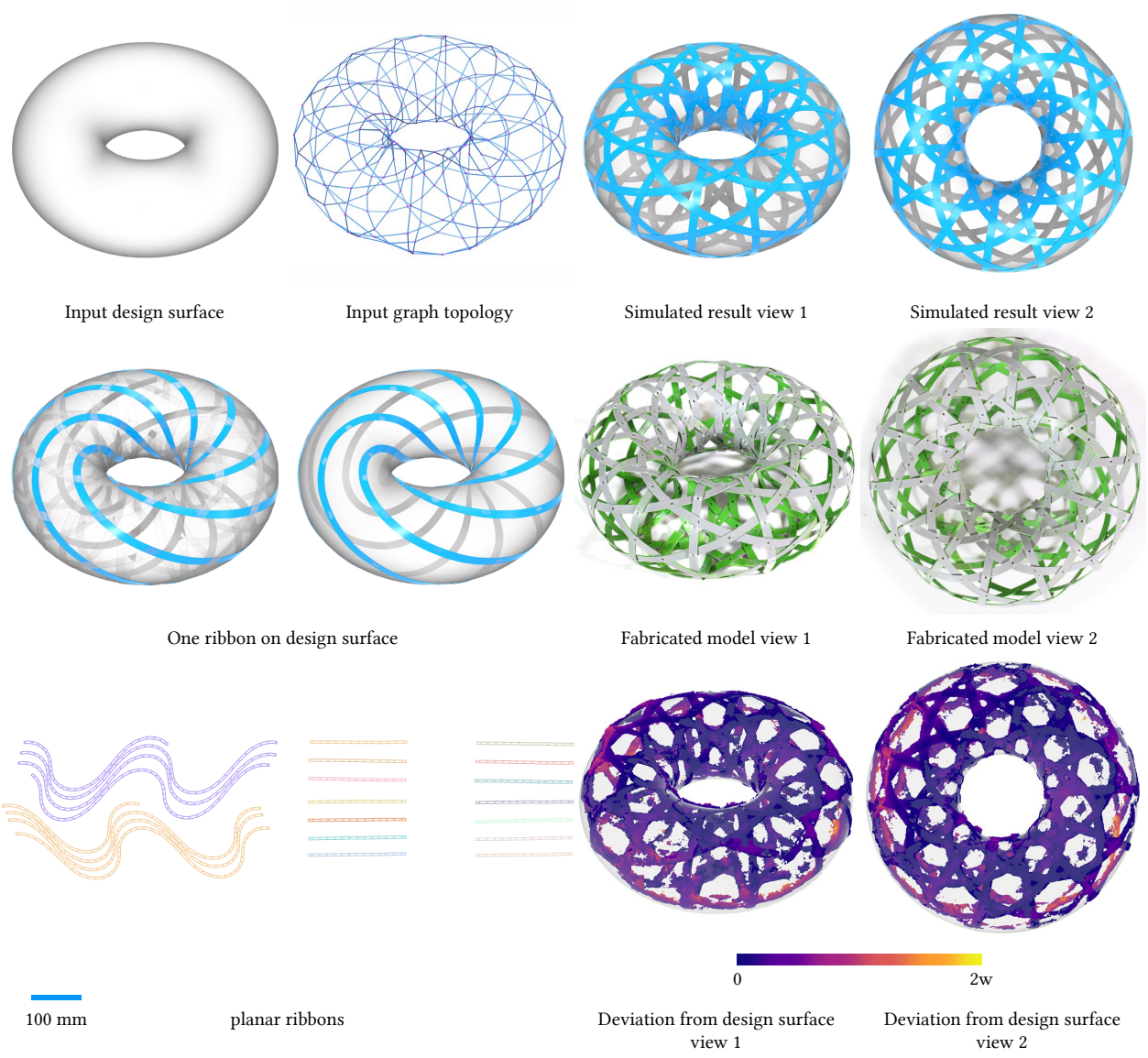

Figure 2 *Torus*

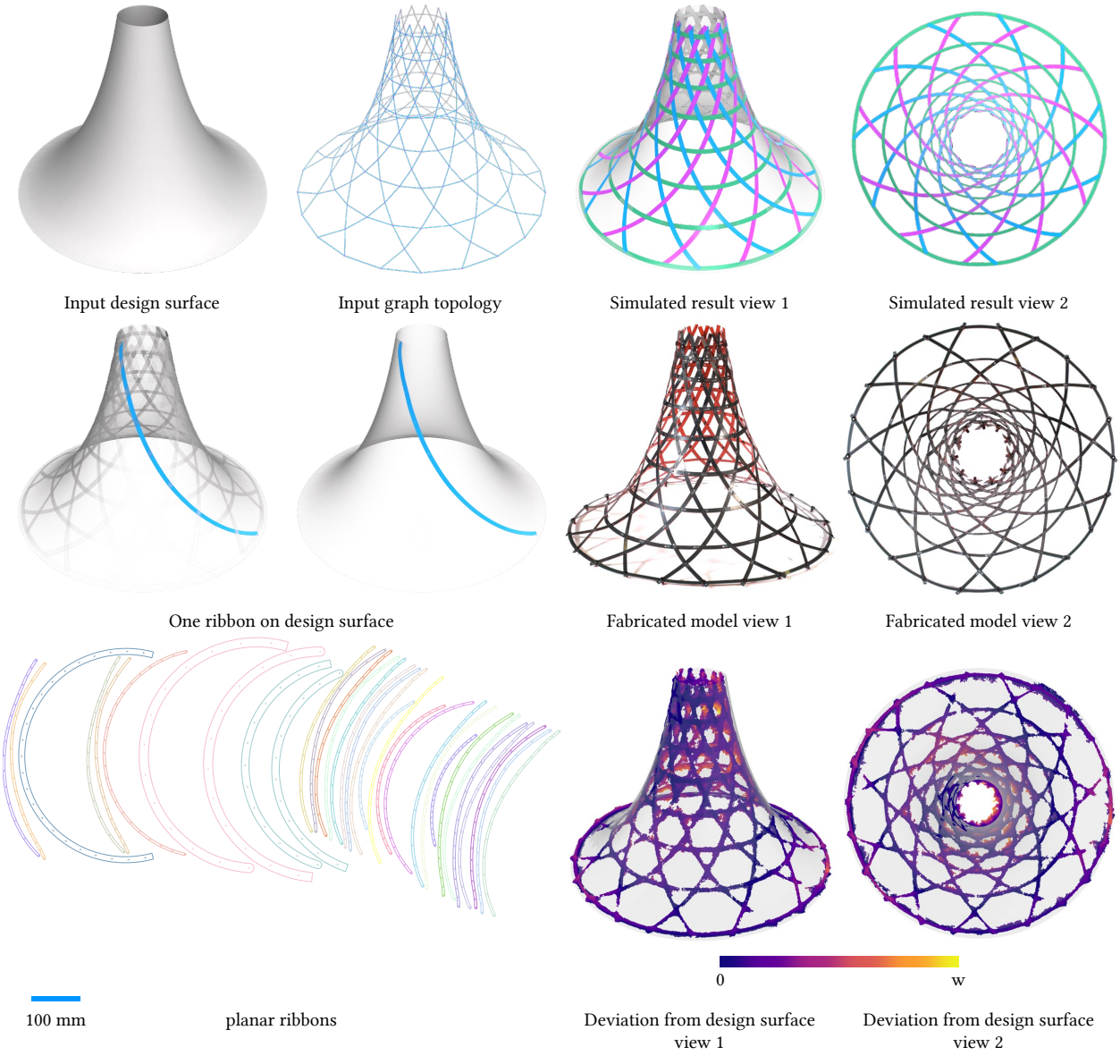

Figure 3 *Pseudosphere*

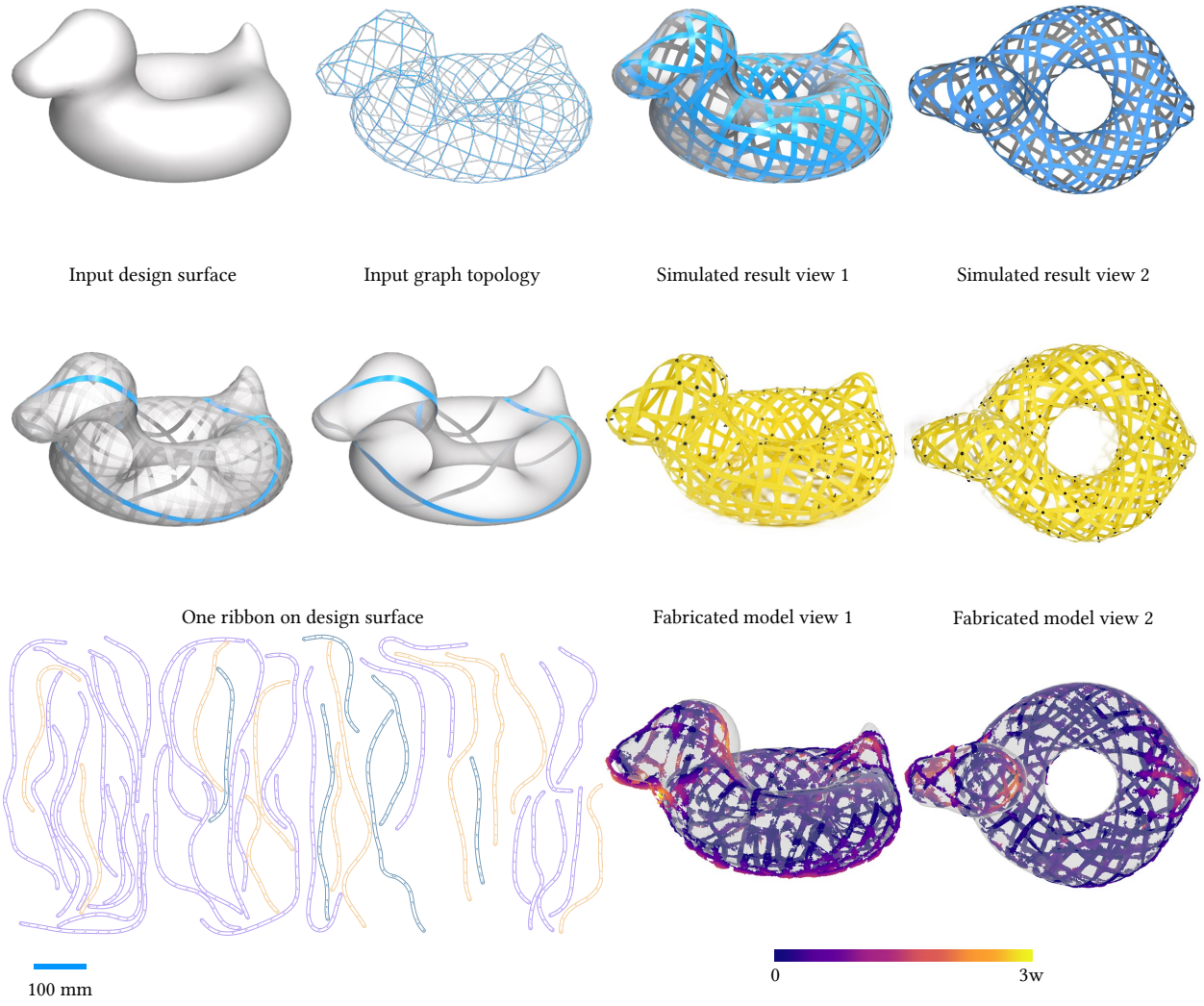

Figure 4 *Duck (15(d)))*

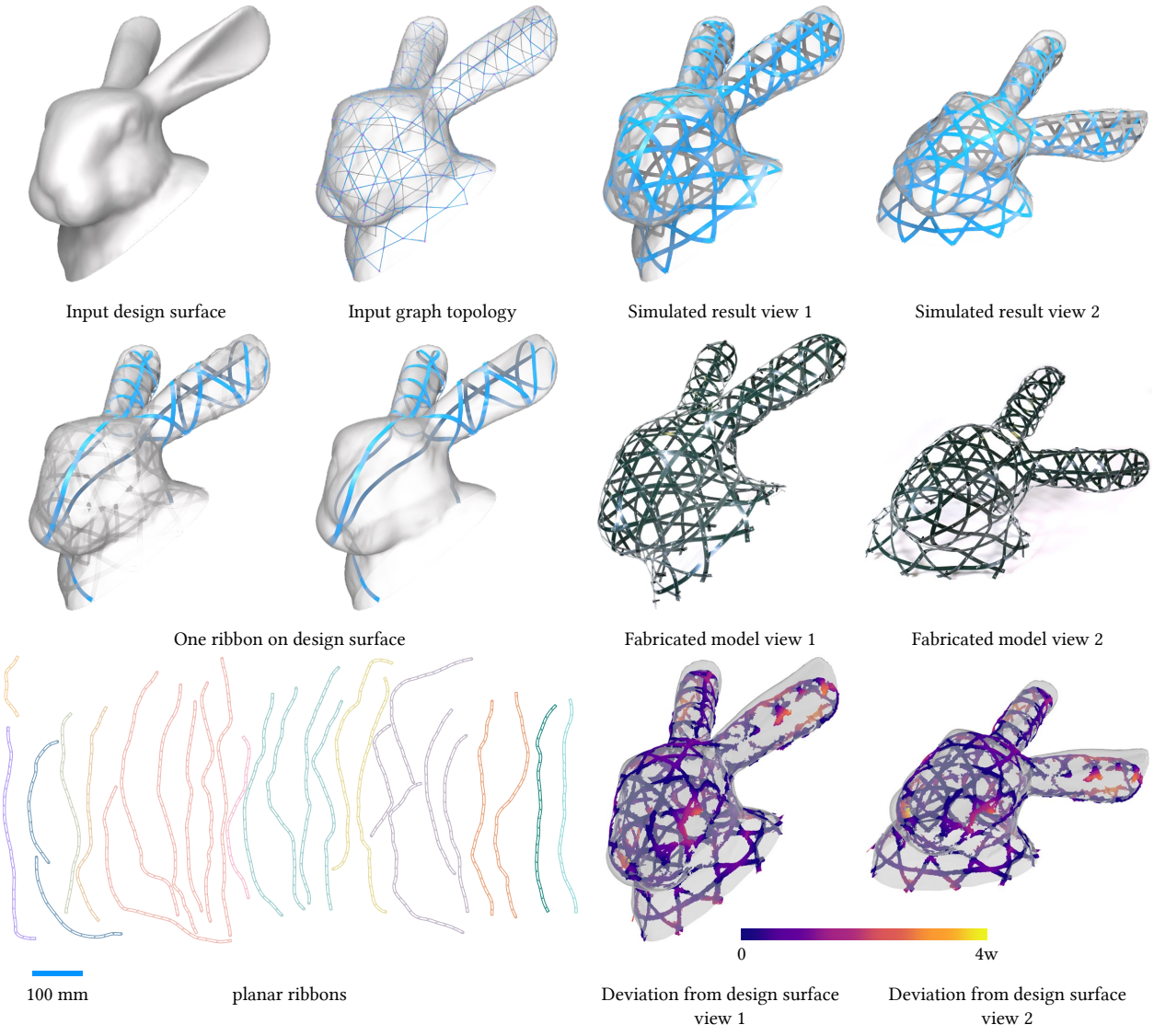

Figure 5 *Bunny*

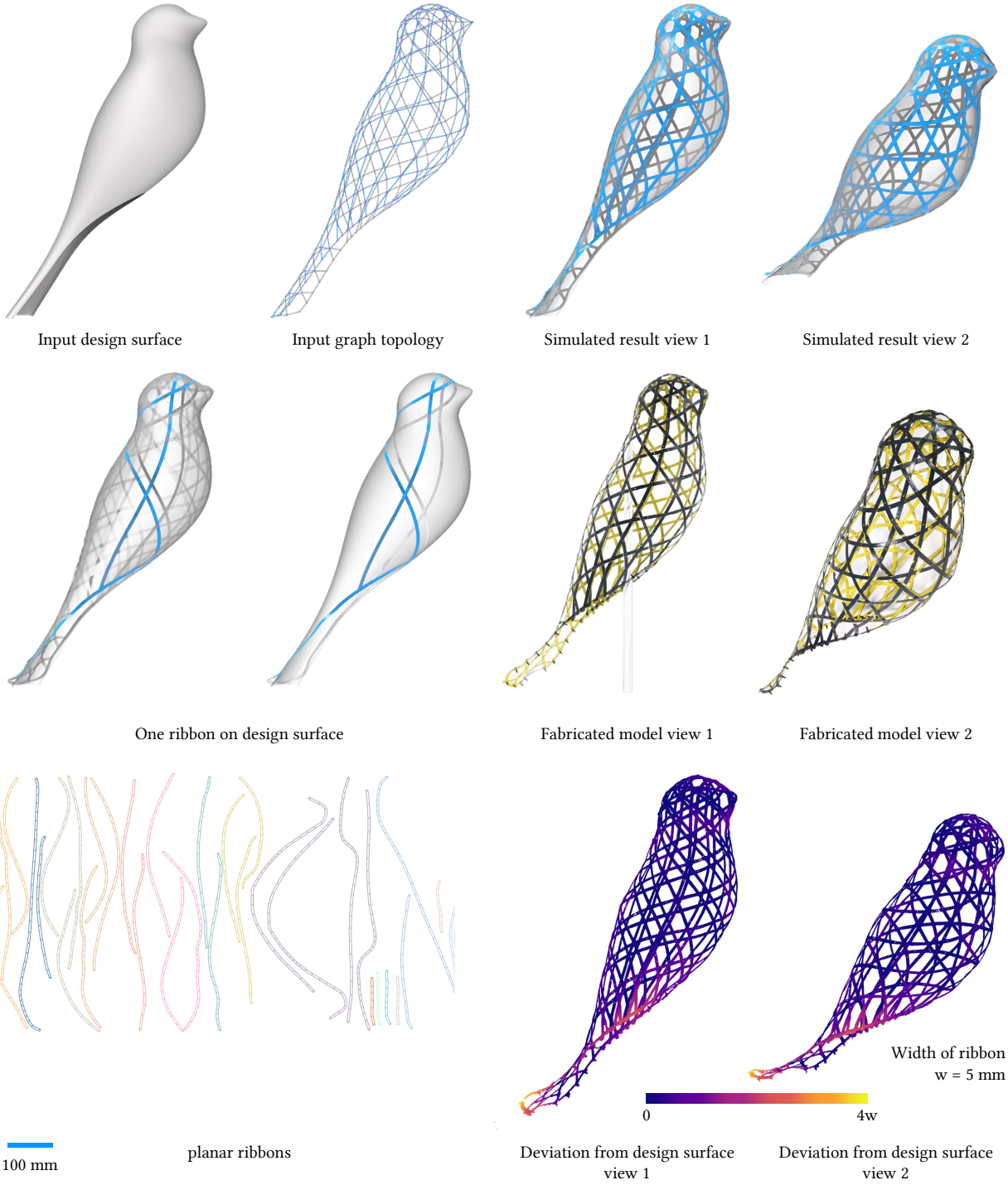

Figure 6 *Bird*

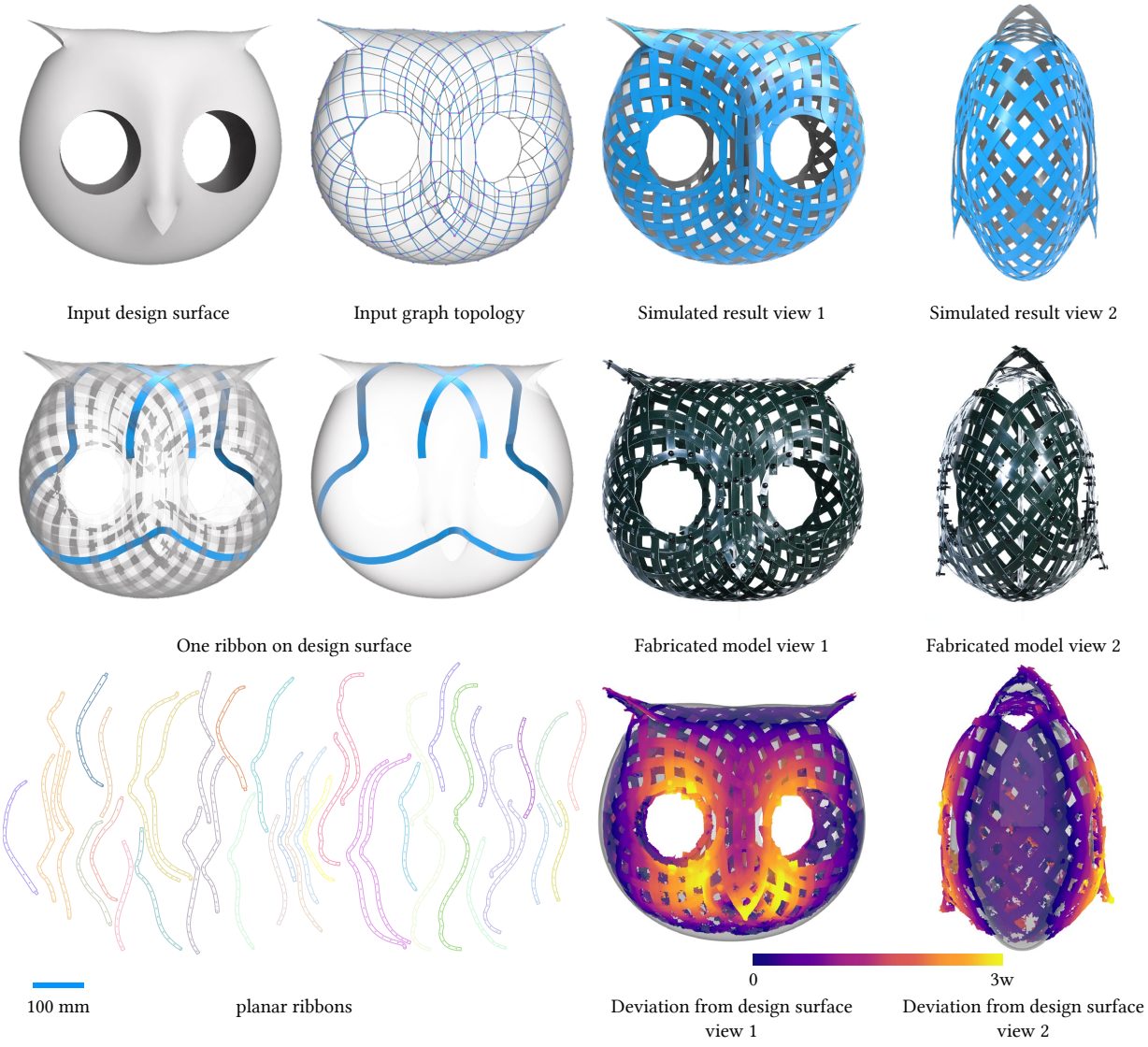
Input design surface
Input graph topology
Simulated result view 1
Simulated result view 2
One ribbon on design surface
Fabricated model view 1
Fabricated model view 2
100 mm
planar ribbons
0
3w
Deviation from design surface view 1
Deviation from design surface view 2

Figure 7 *Owl*

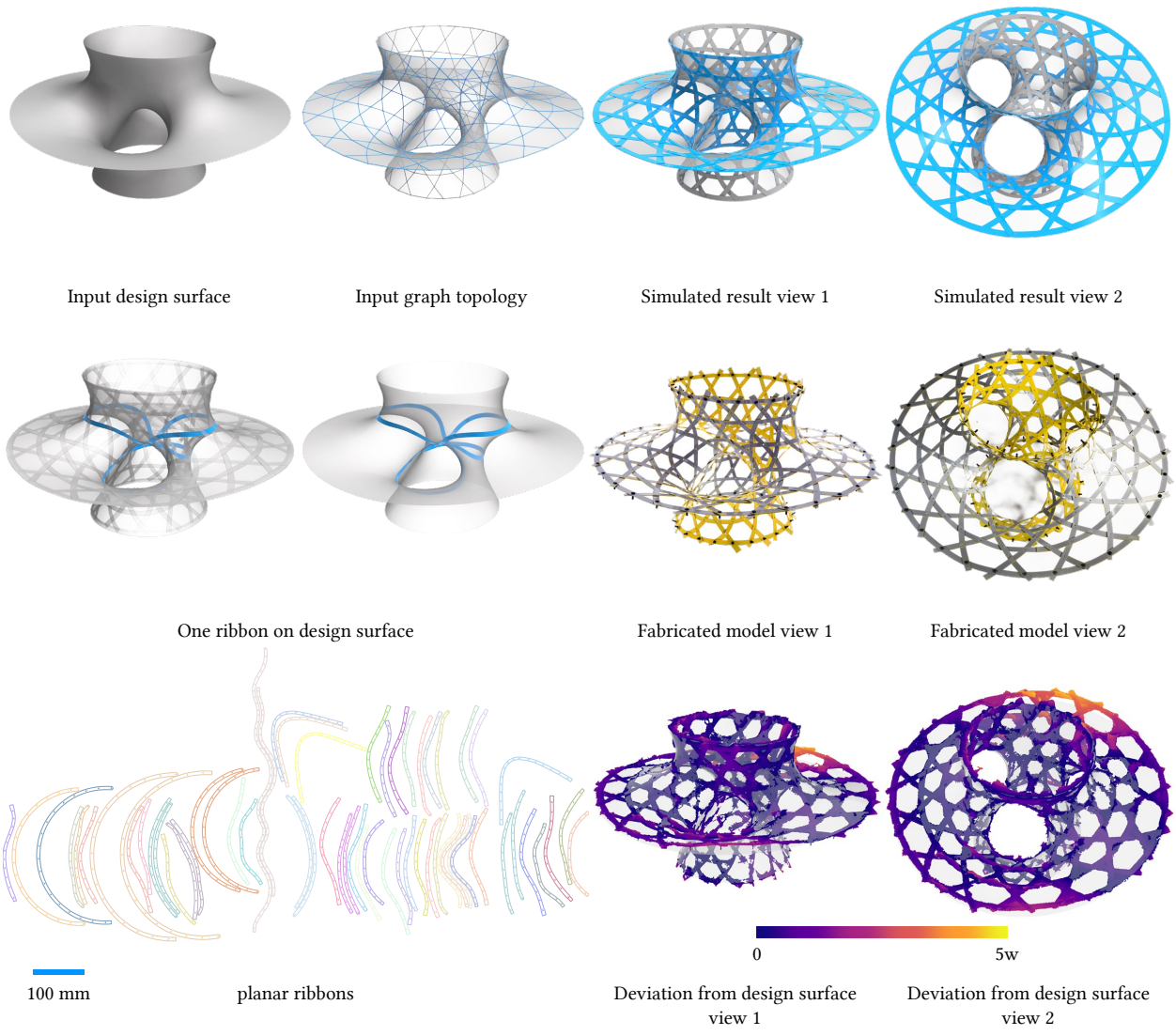

Figure 8 *Costa*

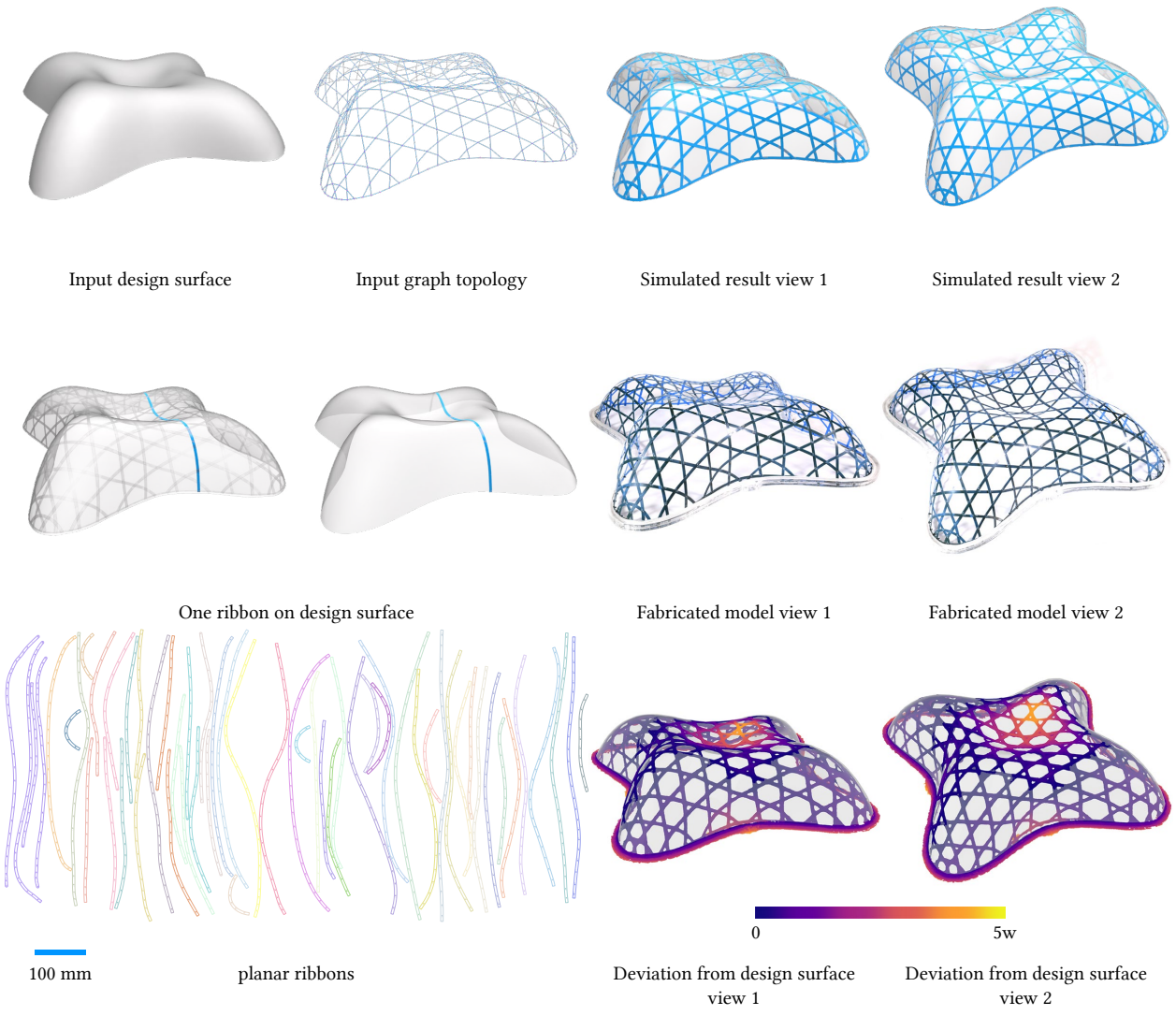

Figure 9 *Lilium*

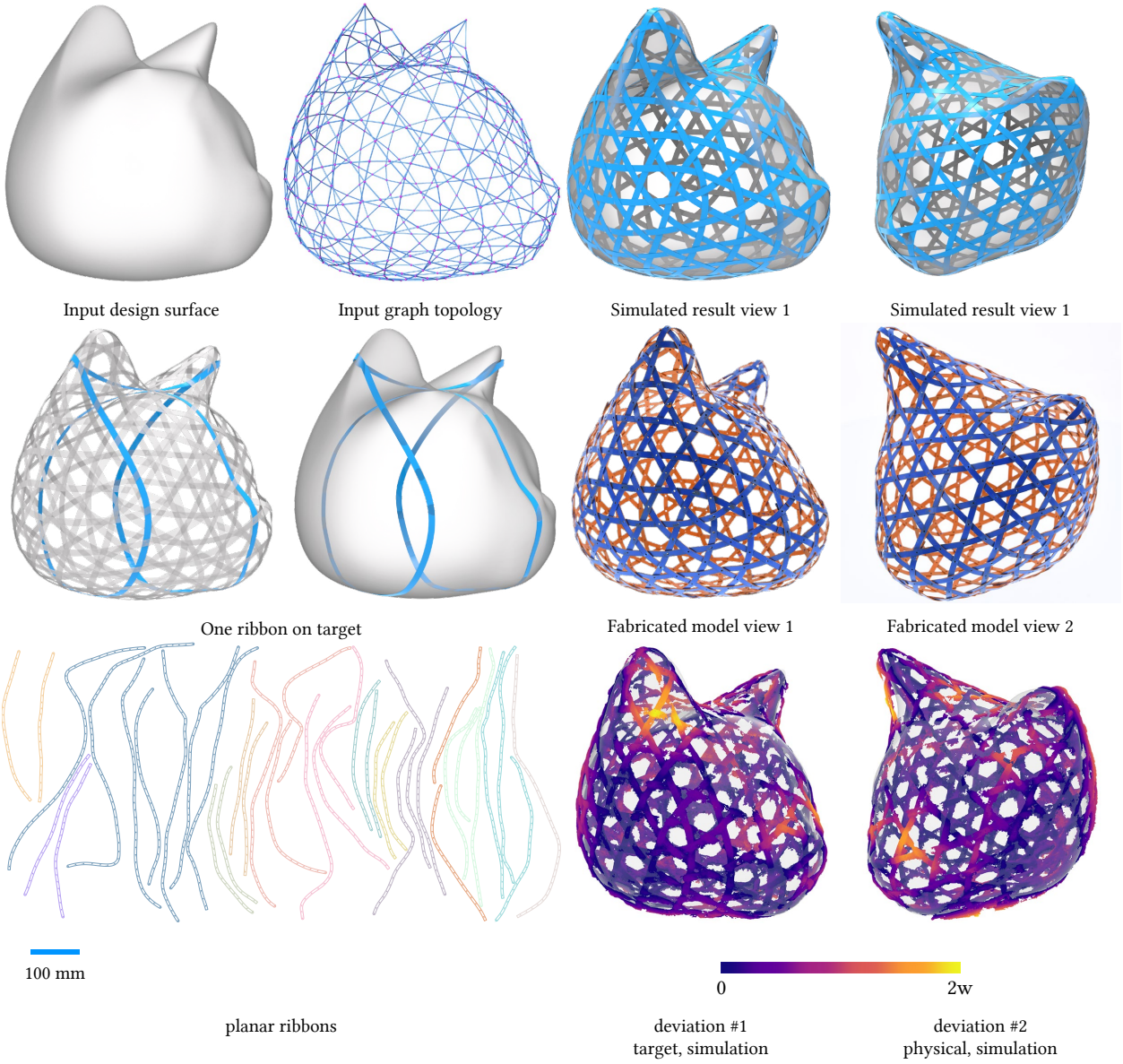

Figure 10 *Cat A*

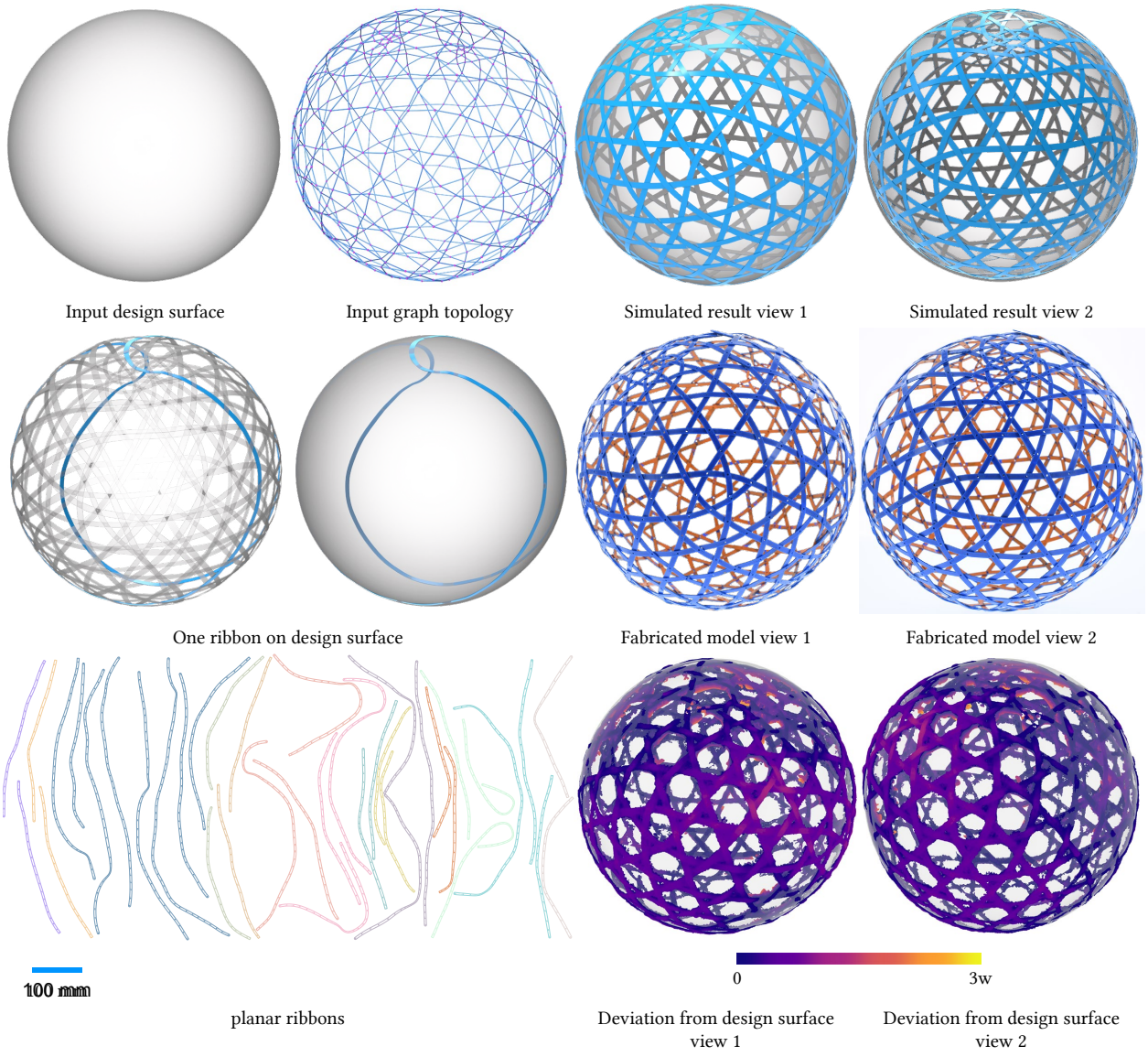

Figure 11 *Cat E*

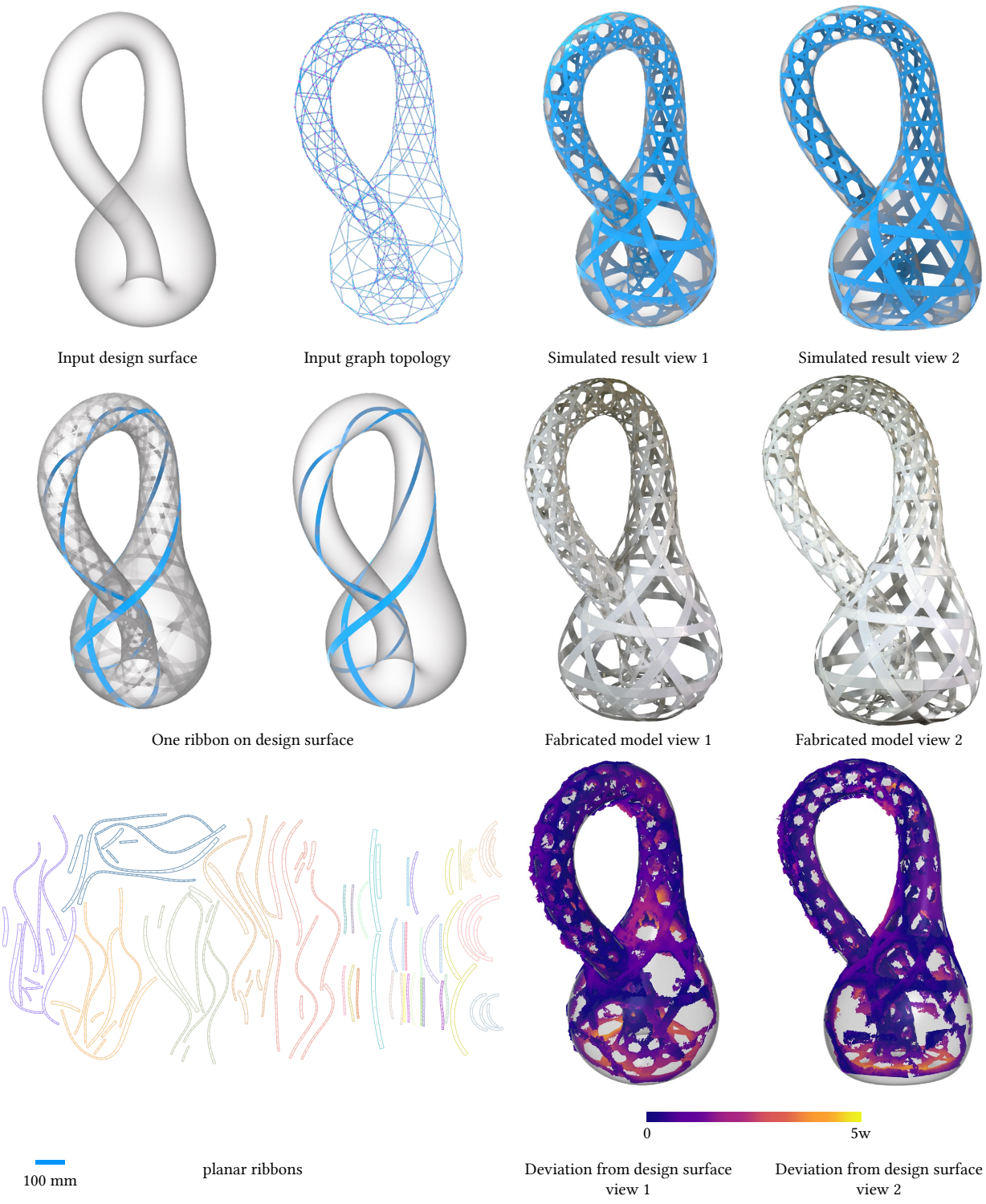

Figure 12 *Klein Bottle*