Bayesian Differentiable Physics for Cloth Digitalization

Supplementary Material

1. Cusick Drape Dataset

Our current Cusick drape dataset includes 25 types of common fabrics, each of which with multiple samples. As listed in Tab. 1, these fabrics are different in material, woven pattern, area density (ρ), and average thickness. In our Cusick drape test, they also show distinctive drape shapes. We test each sample multiple times. In every test, our Cusick drape meter captures a drape image and reconstructs its 3D mesh. It took approximately more than 15 working days to do the textile testing. At the end, there are 660 drape images and meshes in our current dataset which will be released with our paper. Fabric 1-5 are used in our experiments which are the Cotton White, Cotton Blue, Viscose White, Cotton Pink, and Wool Red respectively.

The lack of training data is the deciding factor that hinders the application of deep learning in textile. To our best knowledge, the only open source cloth drape dataset was proposed in [4] very recently. Compared with their dataset, our new dataset has multiples advantages. First, our dataset includes 25 types of common fabrics and has more training samples, which comprehensively covers a wide range of fabrics, with accurate description of the materials. The number of types of fabrics and details are unclear in their data. Second, their data only provides the estimated material parameters tied to their estimation methods, which might make it difficult to transfer them to a different cloth model. In contrast, not only do we provide the raw Cusick drape testing data (*i.e.* images and meshes), we also provide the estimated parameters. Furthermore, their parameters have specific values while ours are distributions that capture cloth material heterogeneity and dynamics stochasticity. Last but not the least, our Cusick drape testing rigidly follows the British standards and the textile measurements (e.g. measuring area density and thickness) are conducted in a rigorously controlled environment so that they are more accurate and easier to be reproduced/verified in future research. By contrast, their drape testing is measured by customized apparatus under a less controlled setting.

2. Performance

Simulation efficiency is critical to our cloth digitalization method because (1) training BDP usually requires longer time than deterministic models, and (2) generalizing the digitialized cloths on garments needs to simulate larger and more complex meshes. Our implementation of the differentiable cloth model is highly inspired by [5]. But ours is more GPU-friendly, and therefore runs fast. We compare the forward simulation and the backward gradient computation, which are two major time-consuming operations for differentiable physics models, between our implementation and [5]. We use three meshes with different resolutions, consisting of 279, 1205, and 2699 vertices respectively. Tab. 2 shows the significant performance gain by our vectorization and GPU parallel computing. The test is conducted on a PC with an Intel Xeon E5-1650 v4 3.60GHz CPU and an NVIDIA TITAN Xp GPU.

3. Additional Results

3.1. Necessity of Material Heterogeneity

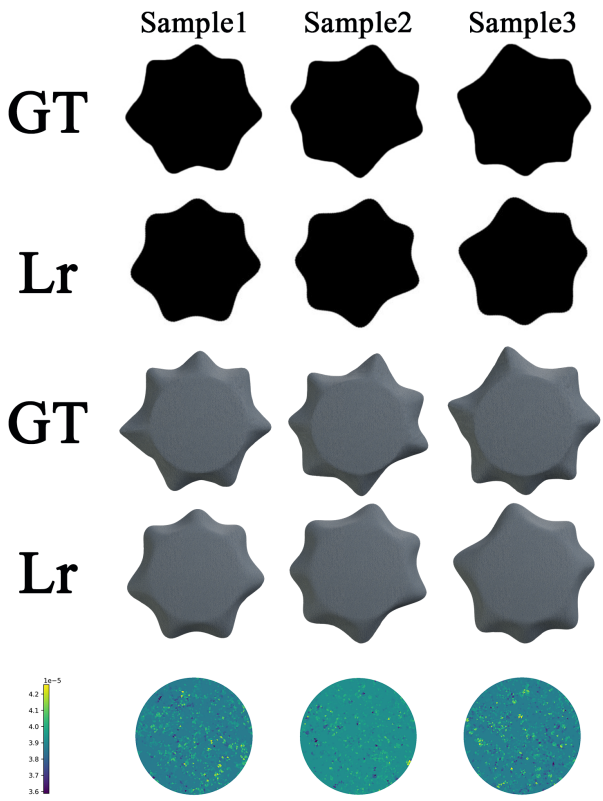


Figure 1. Learning from drape shapes of the 3 samples cut from fabric Cotton Blue. The GT rows show the within-sample variations in drape shapes. The LR rows show the HETER model accurately learns these different drape shapes. The heat maps (bottom) show distributions the per-element bending stiffness $\mathbf{B}$ over the mesh.

Our model attributes the within-sample shape variation to cloth material heterogeneity. To demonstrate it, we use the HETER (deterministic heterogeneous model) to learn

Fabric Index	Material	Woven	# Samples	Avg ρ (kg/m^2)	Avg Thickness(mm)
Fabric 1	Cotton	Plain	12	0.059	0.188
Fabric 2	Cotton	Basket	12	0.192	0.402
Fabric 3	Viscose(95%) Elastane (5%)	Knit	12	0.213	0.560
Fabric 4	Cotton	Plain	12	0.114	0.200
Fabric 5	Wool	Twill	12	0.274	0.571
Fabric 6	Polyester	Satin	12	0.183	0.240
Fabric 7	Polyester(65%) Cotton(35%)	Plain	12	0.100	0.195
Fabric 8	Linen	Plain	12	0.230	0.485
Fabric 9	Cotton	Plain	12	0.249	0.423
Fabric 10	Viscose (70%) Polyester (30%)	Plain	12	0.204	0.498
Fabric 11	Wool	Twill	2	0.148	0.210
Fabric 12	Cotton	Plain	2	0.313	0.188
Fabric 13	Cotton	Plain	2	0.107	0.15
Fabric 14	Cotton	Plain	2	0.139	0.292
Fabric 15	Cotton	Plain	2	0.211	0.436
Fabric 16	Synthetic	Knit	2	0.191	0.514
Fabric 17	Cotton	Twill	2	0.278	0.7
Fabric 18	Synthetic	Plain/Knit	2	0.154	0.306
Fabric 19	Synthetic	Plain/Knit	2	0.282	0.627
Fabric 20	Synthetic	Twill	2	0.259	0.596
Fabric 21	Synthetic	Knit	5	0.186	0.7
Fabric 22	Synthetic	Plain	5	0.231	0.386
Fabric 23	Cotton	Knit	5	0.163	0.704
Fabric 24	Cotton	Plain	5	0.104	0.200
Fabric 25	Cotton	Plain	5	0.060	0.152

Table 1. Fabric information in our Cusick drape dataset. Fabric 18 and Fabric 19 are double-layer fabrics whose one side is plain woven and the other side is knit.

Test (sec/step)	Mesh 1	Mesh 2	Mesh 3
[5] Forward	0.643	2.793	3.861
[5] Backward	1.400	7.379	26.212
Ours Forward	0.038	0.073	0.178
Ours Backward	0.062	0.228	0.660

Table 2. Comparing the run time between [5] (collision handling off) and our model in forward simulation and backward gradient computation. Our implementation is much faster.

the drape shapes of six samples (three from Cotton Blue and three from Viscose White). The ground-truth (GT) rows in Fig. 1 and Fig. 2 show that samples from the same fabric are obviously different. The learned (LR) rows in Fig. 1 and Fig. 2 demonstrate that the HETER can accurately learn from the different drape shapes. The heat maps in the last row illustrate the learned distributions of the cloth physical parameters across the meshes. This result demonstrates that within-sample drape shape variation can be accounted for by cloth material heterogeneity and varied physical parameter distributed across the mesh in different samples.

3.2. Learning from Meshes

We train the model using the reconstructed 3D meshes as the ground-truth which include geometry information, by minimizing the MSE between the simulated and real meshes. As shown in Fig. 3, our BDP also exhibits outstanding training sample fitting and generalization ability. Quantitatively, the MSE and H.Dis are 0.051 and 0.023 under the same setting as Table 1 in the paper. Comparatively, 3D meshes do not improve the results and most testers cannot generate 3D mesh data.

3.3. Effectiveness of Learning

To digitalize cloths, our BDP uses gradient-based optimization to find cloth physical parameters such that cloth drape shape can be optimized toward the given ground truth sample. Fig. 4 shows that the optimization is effective, where the initial guess on the parameters lead to initial drape shapes that are obviously different from the GT, but are adjusted toward the GT after training.

The digitalized cloths can be applied into garment simulation and obviously reflect different cloth mechanical char-

Cloths	Cotton White	Cotton Blue	Viscose White	Cotton Pink	Wool Red
Avg. Stretching	56.6019	63.9698	4.5117	101.4625	78.3332
Avg. Bending	1.008×10^{-5}	8.249×10^{-5}	4.991×10^{-5}	4.829×10^{-5}	0.0001
Avg. Std	1.435×10^{-6}	0.0001	0.0214	3.860×10^{-6}	2.057×10^{-5}

Table 3. The average stretching and bending stiffness of the five select cloths reflect their distinguishable physical properties. In addition, their average physical parameter standard deviations show their different material heterogeneity and dynamics stochasticity.

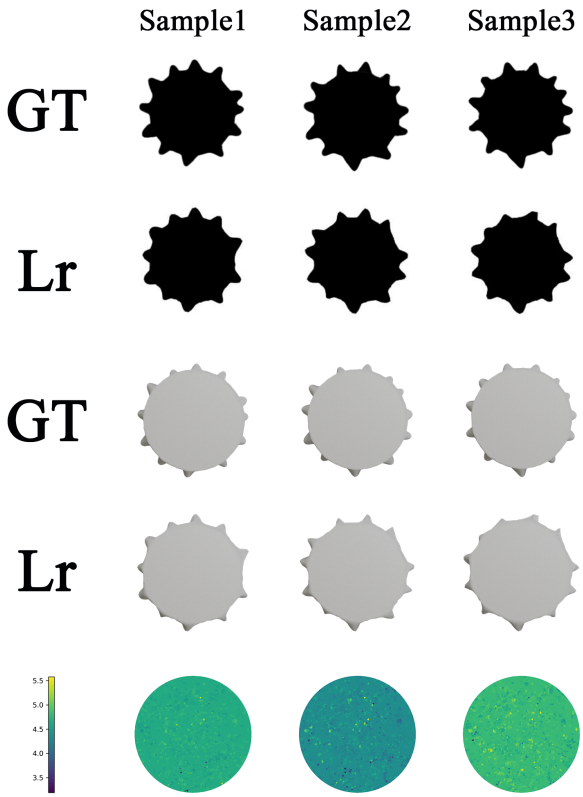


Figure 2. Learning from drape shapes of the 3 samples cut from fabric Viscose White. The GT rows show the within-sample varied draped shapes. The LR rows show the HETER model accurately learns these different drape shapes. The heat maps (bottom) show distributions the per-element stretching stiffness $\mathbf{C}$ over the mesh.

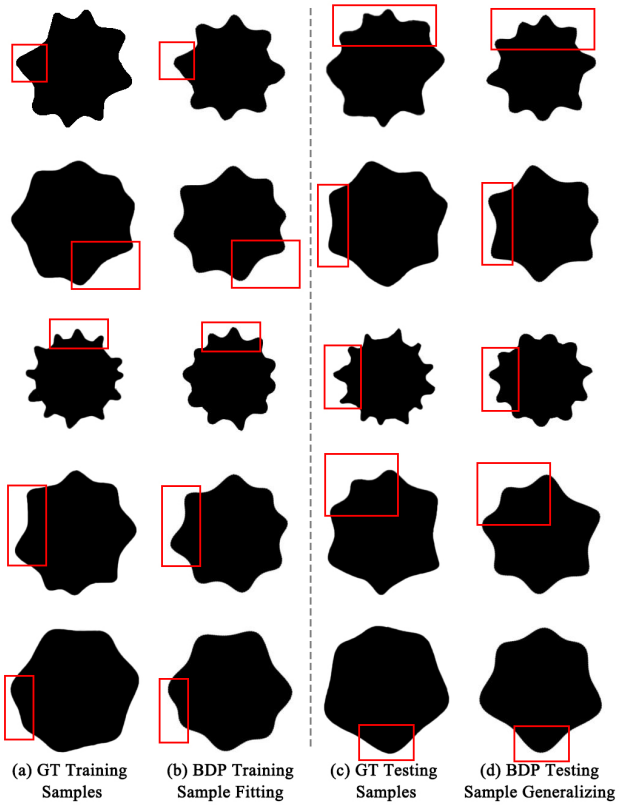


Figure 3. When learning from meshes, BDP can also fit the training sample(a) and generalize to the unseen testing samples(c).

acteristics in various motions (as shown in Figs. 5 to 7). For example, in Fig. 6, there are more folds on the Viscose White than on the Wool Red, because the former is much softer (small bending stiffness) than the latter.

In addition, our BDP embeds cloth material heterogeneity and dynamics stochasticity (Fig. 8). Conversely, the HOMO is unable to digitalize and simulate these cloth properties (Fig. 9).

4. More comparison

Fig. 10 provides the visual comparison between the gradient-based (our BDP) and the gradient-free optimization, *i.e*. Bayesian Optimization (BO), method. It confirms the quantitative result given in the main paper that the BDP gains a better results with fewer optimization steps than the BO.

5. Posterior Space

Although we mainly focus on digitalizing cloths, the learned posterior space has other potentials. The first one is doing a quantified comparison between different cloth types. The five digitialized cloths in our experiment are

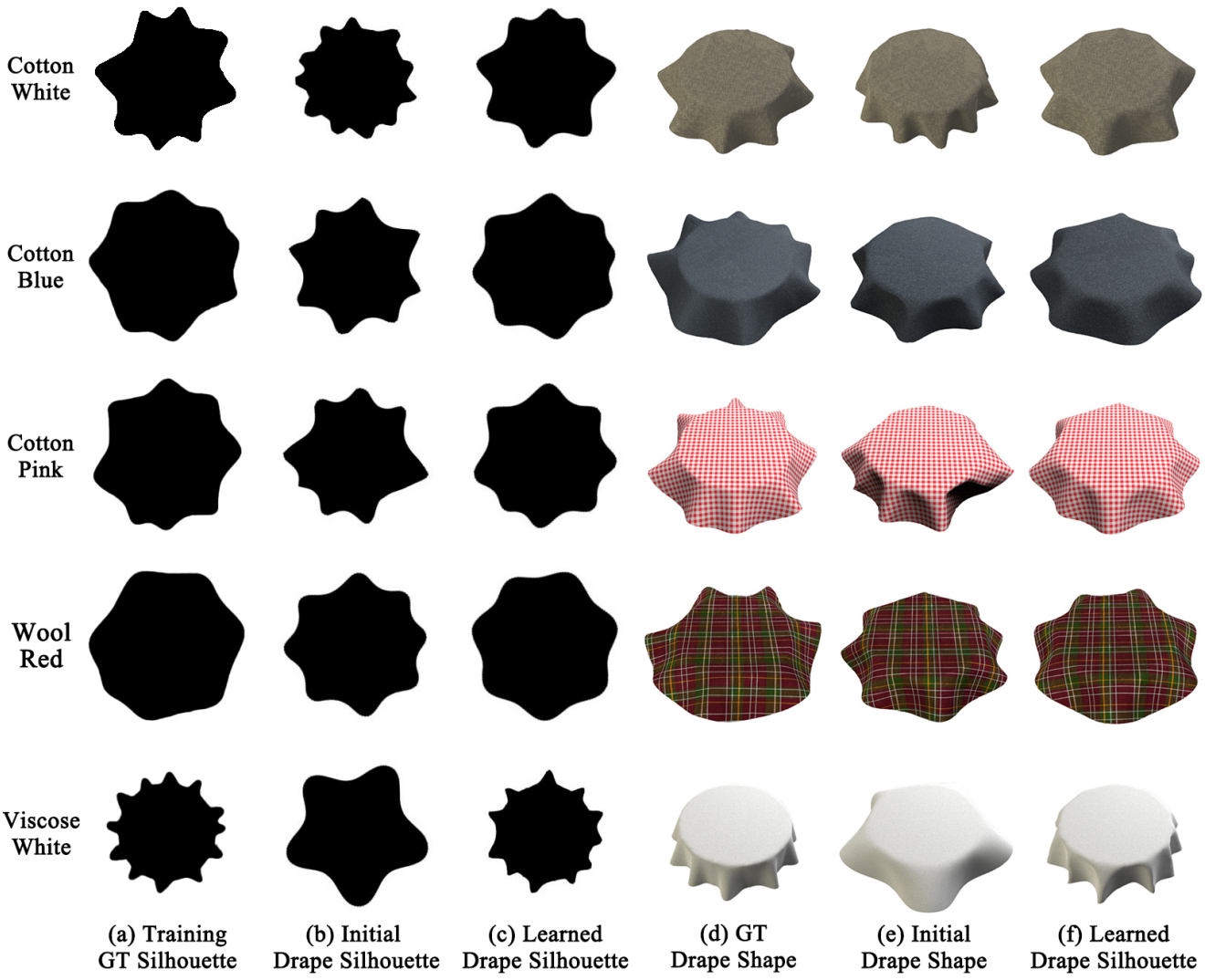


Figure 4. Given the initial parameters, the simulated drape shapes (b and e) are different from the training samples. After training, our BDP optimize cloth physical parameters and fit the given GT drape silhouettes (c and f).

representative, where we can derive some insights from the learned parameter distributions. As shown in Tab. 3, the estimated parameters also reflect our daily qualitative observations of the cloths. For instance, the mean of the Viscose White's stretching stiffness parameters are smaller than the other cloths' because it is a knitted fabric which is soft and can be used to make T-shirt or socks which are very stretchable. By contrast, the mean of the Cotton Pink's stretching stiffness parameters are larger because it is a tightly woven fabrics which is stiff, and can be used as tablecloths. In addition, the standard deviation of the Viscose White's physical parameters are larger because its loose woven pattern tends to cause material heterogeneity and lead to dynamics stochasticity. This kind of analysis has been pursued so far in textile when it comes to Cusick drape data, but rarely with the kind of material details (reflected in *e.g.* material heterogeneity and dynamics stochasticity) given by our BDP model.

Furthermore, enabled by BDP, we could learn all cloth types in our dataset or simply learn a subset of representative ones, then combine the learned parameter distributions into a Gaussian Mixture Model (GMM) to represent the material space of all common fabrics. This GMM could then be employed for other applications such as detecting whether a tested material is likely to be a common fabric, or even quality control in detecting anomaly material *e.g.* due to manufacturing defects. This could be done by first learning the material, then compute the likelihood of the es-

(a) Cotton White (b) Viscose White (c) Wool Red (d) Cotton Blue (e) Cotton Pink

Figure 5. The digitalized cloths exhibit different mechanical characteristics in the walking motion.

(a) Cotton White (b) Viscose White (c) Wool Red (d) Cotton Blue (e) Cotton Pink

Figure 6. The digitalized cloths exhibit different mechanical characteristics in the dancing motion.

timated material parameters in the GMM. Although this is way out of the scope of this paper, we can still give a simple

Figure 7. The digitalized cloths exhibit different mechanical characteristics in the jumping motion.

Cloths	Cotton White	Cotton Blue	Viscose White	Cotton Pink	Wool Red	Sum
Linen Green	8.669×10^{30}	1.202×10^{19}	9.796×10^{30}	7.307×10^{30}	7.707×10^{29}	2.654×10^{31}
Paper	2.253×10^{38}	2.253×10^{38}	2.254×10^{38}	2.252×10^{38}	2.253×10^{38}	1.126×10^{39}

Table 4. The Kullback–Leibler(KL) divergence of the estimated parameters from the Linen Green sample and the paper parameters to our selected five representative cloths' parameters probability distributions. The smaller the KL divergence is, the closer the parameters are to the distributions. Thus, (1) the "Linen Green" is more like Cotton Blue among the five kinds of cloths and (2) the "Linen Green" is more like fabric than paper.

illustrative example. We select a sample from a cloth type that is different from the five types for the learning so far, refer to it as "Linen Green", and digitalize it by our BDP model (Fig. 11 (a-c)). To make the comparison more visual, we make a paper-like material which shows obviously different drape shape (as shown in Fig. 11 (e)). Then we compare the likelihoods of their material parameters in the GMM consisting of the five Gaussians learned in our experiments. However, both likelihoods are very low, despite the likelihood of Linen Green is several magnitudes higher than paper-like. This suggests that only five types are not sufficient to capture the full common cloth material space. Learning more cloth types in our dataset will mitigate this issue. But we can still compute similarities between the newly learned distributions and the original five types, measured by the Kullback-Leibler (KL) Divergence . The results are shown in Tab. 4. Interestingly, Linen Green is by far most similar to Cotton Blue compared with other types, which is consistent with qualitative observations (as shown in Fig. 11 (d)). In contrast, paper-like is dissimilar to every material, which is expected. In future, we will explore what materials need to be learned and incorporated into this GMM model so that the posterior distribution captures better the full material parameter space.

6. Force Models

In this section, we give more details about our cloth model. A cloth sample is modeled as a circular triangular mesh con-

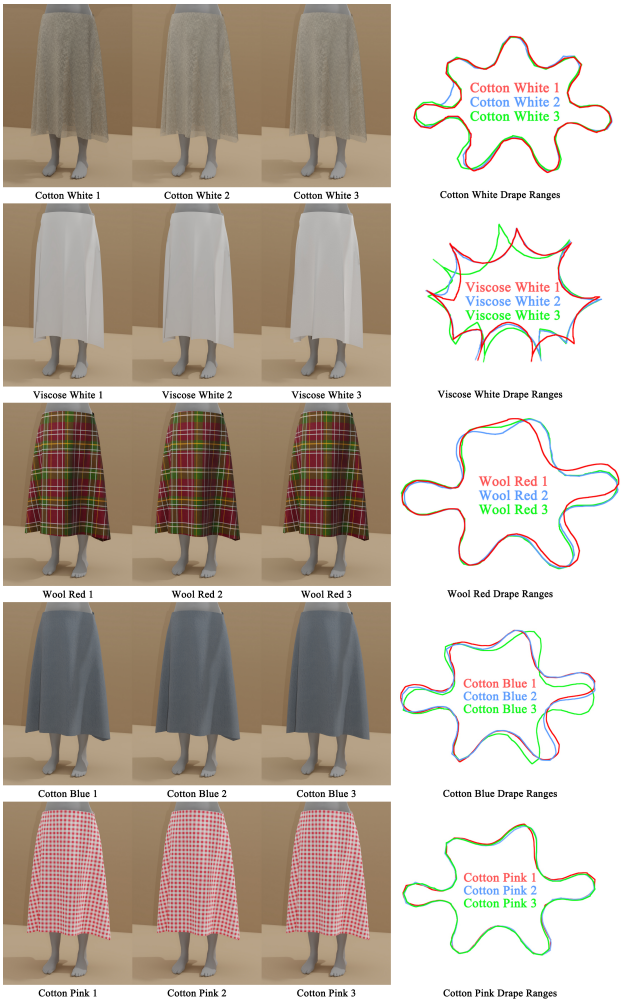


Figure 8. Given the digitialized cloths, our BDP model can simulate the skirts made from these cloths and reflect cloth material heterogeneity and draping stochasticity.

sisting of v vertices. Its state $\mathcal{S}$ is defined by its vertices position, $\mathbf{x} \in \mathbb{R}^{3\times v}$, and velocity, $\dot{\mathbf{x}} \in \mathbb{R}^{3\times v}$, in Cartesian coordinates. Therefore, a cloth motion is represented by a state sequence: $\mathcal{S}_{0:n} = \{\mathcal{S}_t : t \in \mathbb{Z}^+; t \leq n\}$, which denotes the position and velocity change over time. A physics-based cloth simulator aims to simulate a cloth motion by recurrently predicting its future state $\mathcal{S}_{t+1} = \{\mathbf{x}_{t+1}, \dot{\mathbf{x}}_{t+1}\}$ given the current state $\mathcal{S}_t = \{\mathbf{x}_t, \dot{\mathbf{x}}_t\}$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h\dot{\mathbf{x}}_t \tag{1}$$
$$\dot{\mathbf{x}}_{t+1} = \dot{\mathbf{x}}_t + h\ddot{\mathbf{x}}_t \tag{2}$$

where h is the time step size (time lapse between every two consecutive states) and the second-order time derivative, $\ddot{\mathbf{x}}_t$, is vertices acceleration. To gain high simulation stability, implicit Euler method [2] is commonly used:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h\dot{\mathbf{x}}_{t+1} \tag{3}$$
$$\dot{\mathbf{x}}_{t+1} = \dot{\mathbf{x}}_t + h\ddot{\mathbf{x}}_{t+1} \tag{4}$$

According to Newton's Second law, we have

$$\mathbf{F} = \mathbf{M}\mathbf{a} = \mathbf{M}\ddot{\mathbf{x}} \tag{5}$$

where $\mathbf{M}$ is the general mass matrix and $\mathbf{F}$ is the resultant force which is the combination of internal and external forces. In our differentiable cloth simulator, these forces are decided by cloth sample's current state, so we can define:

$$\mathbf{F}_t = \mathbf{f}(\mathcal{S}_t) = \mathbf{f}(\mathbf{x}_t, \dot{\mathbf{x}}_t) \tag{6}$$

where $\mathbf{f}$ denotes a general function, which must be differentiable, takes as input the current state, $\mathcal{S}_t$, and outputs the resultant force. Through Taylor approximation[2], Eq. (3) and Eq. (4) are converted to the governing equation of the physical system:

$$\left(\mathbf{M} - h\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}} - h^2\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Delta\dot{\mathbf{x}} = h\left(\mathbf{F}_t + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\dot{\mathbf{x}}_t\right) \tag{7}$$

which needs to be solved to calculate $\Delta\dot{\mathbf{x}}$ and update the cloth sample's state:

$$\dot{\mathbf{x}}_{t+1} = \dot{\mathbf{x}}_t + \Delta\dot{\mathbf{x}} \tag{8}$$
$$\mathbf{x}_{t+1} = \mathbf{x}_t + h\dot{\mathbf{x}}_{t+1} \tag{9}$$

In our differentiable cloth simulator, the resultant force is defined as: $\mathbf{F} = \mathbf{F}_{stretch} + \mathbf{F}_{bend} + \mathbf{F}_{gravity} + \mathbf{F}_{handle}$. The stretching force [8] on a face j is:

$$\mathbf{F}^{(j)}_{stretch} = -A^{(j)}\left(\sum_{m\in(uu,vv,uv)} \sigma^{(j)}_m\left(\frac{\partial \varepsilon^{(j)}_m}{\partial \mathbf{x}_i}\right)\right) \tag{10}$$

where $\varepsilon^{(j)}_m$ denotes stretching strain and the $\mathbf{x}_i$ are the three vertices of the face. The stretching stresses $\sigma^{(j)}_m$ is

$$\begin{bmatrix}\sigma^{(j)}_{uu}\\ \sigma^{(j)}_{vv}\\ \sigma^{(j)}_{uv}\end{bmatrix} = \begin{bmatrix}c^{(j)}_{11} & c^{(j)}_{12} & 0\\ c^{(j)}_{12} & c^{(j)}_{22} & 0\\ 0 & 0 & c^{(j)}_{33}\end{bmatrix}\begin{bmatrix}\varepsilon^{(j)}_{uu}\\ \varepsilon^{(j)}_{vv}\\ \varepsilon^{(j)}_{uv}\end{bmatrix} = \mathbf{C}^{(j)}\boldsymbol{\varepsilon}^{(j)} \tag{11}$$

where $c^{(j)}_{11}$, $c^{(j)}_{12}$, $c^{(j)}_{22}$, and $c^{(j)}_{33}$ are the stretching stiffness in weft/course direction, the stretching stiffness in warp/wale direction, Poisson's ratio, and shearing stiffness. The subscripts uu, vv, and uv denote cloths weft/course, warp/wale, and diagonal directions respectively (Fig. 12 Left). $\boldsymbol{\varepsilon} = [\varepsilon^{(j)}_{uu}, \varepsilon^{(j)}_{vv}, \varepsilon^{(j)}_{uv}]$ is the Voige form strain tensor. The bending force around a bending edge w is defined as

$$\mathbf{F}^{(w)}_{bend} = \mathbf{B}^{(w)}\frac{|\mathbf{e}^{(w)}|}{\psi^{(w)}_1 + \psi^{(w)}_2}\sin\left(\frac{\gamma^{(w)}}{2} - \frac{\bar{\gamma}^{(w)}}{2}\right)u_i \tag{12}$$

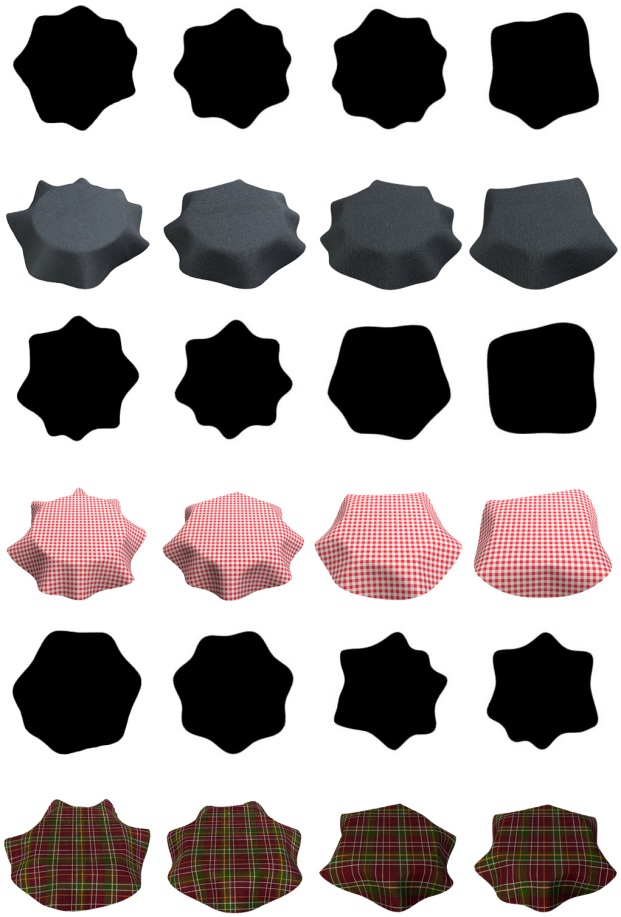
Figure 9. The clothes digitalized by the HOMO model is deterministic and has homogeneous material properties. Thus, the simulated dresses that are made from the same cloth always have the identical geometry.

where $\mathbf{B}^{(w)}$ is the bending edge w's bending stiffness, $|\mathbf{e}^{(w)}|$ is bending edge's rest length, $\psi_1^{(w)}$ and $\psi_2^{(w)}$ are the heights of the two adjacent triangular face, $\gamma^{(w)}$ and $\bar{\gamma^{(w)}}$ is the current and predefined rest dihedral angles between the edge's two adjacent faces (Fig. 12 Middle).

Material non-linearity means the material stiffness changes with deformation magnitude non-linearly. Anisotropy refers to the varied material stiffness in different deformation directions. To encode cloth material non-linearity and anisotropy, our model adopts the piecewise linear physical models in [9] where the stretching stiffness and bending stiffness are defined as two high-dimensional matrices: $\mathbf{C} \in \mathbb{R}^{6\times4}$ and $\mathbf{B} \in \mathbb{R}^{3\times5}$. Then, local stretching stiffness and bending stiffness are sampled from $\mathbf{C}$ and $\mathbf{B}$ according to the mesh's local deformation and geometry. Wang et al.[9] model cloths as continuum elastic shells so a stretching deformation can be described by the Green-Lagrangian strain tensor [3], which can be re-parameterized by Eigen Decomposition:

$$2\begin{bmatrix}\varepsilon_{uu}^{(j)} & \varepsilon_{uv}^{(j)} \\ \varepsilon_{uv}^{(j)} & \varepsilon_{vv}^{(j)}\end{bmatrix} = (\mathbf{R}_{\varphi}^{(j)})^{\top}\begin{bmatrix}(\lambda_{max}^{(j)}+1)^2-1 & 0 \\ 0 & (\lambda_{min}^{(j)}+1)^2-1\end{bmatrix}\mathbf{R}_{\varphi}^{(j)} \tag{13}$$

where the eigenvalues indicate the stretching deformation magnitude and $\mathbf{R}_{\varphi}^{(j)}$ is a rotation matrix that indicates the direction of the stretching deformation. $\lambda_{min}^{(j)}$ can be ignored because they find it has less influence on the stretching stiffness. The rotation matrix is decided by the bias angle, φ, between a cloth sample's rest warp-weft coordinate system and its deformed local coordinate system [7]. This way, the stretching non-linearity and anisotropy can be encoded as the stiffness which changes with parameters in the 2D space spanned by $\lambda_{max}^{(j)}$ and φ. As in [9], we sample 6 data points (the 6 rows in the matrix $\mathbf{C}$) in the polar space



(a) GT (b) BDP (c) REMBO (d) HeSBO

Figure 10. Comparison of ground truth draped shape and the simulated drape shapes learned by ours (b)(derivative-based), REMBO (c) and HeSBO (d)(derivative-free) optimization.

spanned by λ_{max} and φ where each data point contains c_{11}, c_{12}, c_{22}, and c_{33} which compose the 4 columns in the matrix $\mathbf{C}$. (We ignore the face index superscript, (j), to denote the general form.)

To model non-linear bending stiffness, the variables in Eq. (12) can represent the bending deformation so we define a parameter α:

$$\alpha^{(w)} = \frac{\sin(\frac{\gamma^{(w)}}{2} - \frac{\bar{\gamma}^{(w)}}{2})}{h_1^{(w)} + h_2^{(w)}} \tag{14}$$

which is related to the curvature. To model the bending anisotropy, we define another parameter called bending bias angle, i.e. the angle between a bending edge and cloth warp-weft coordinate system's axes, which indicates bending deformation direction (shown in Fig. 13). Therefore, the bending non-linearity and anisotropy can be encoded as the stiffness which changes with the parameter in a polar space spanned by α and bending bias angle. We sample 5 α's and 3 bending bias angles ($0°$, $45°$, and $90°$) which are the 5 columns and the 3 rows of bending stiffness matrix $\mathbf{B}$.

Finally, to model the material heterogeneity, each mesh face and bending edge are associated with a $\mathbf{C}$ and $\mathbf{B}$. Therefore, for a cloth consisting of f faces and e bending edges, the learnable parameters are f stretching matrices and e bending stiffness matrices (the HETER model in our experiments). To our Bayesian differentiable cloth simulator, each $\mathbf{C}$ and $\mathbf{B}$ are sampled from the variational distribution $q_\theta(\tau)$. Its learnable parameter is the distribution parameter θ of $q_\theta(\tau)$. As we assume $q_\theta(\tau)$ is distributed as a Gaussian, θ consists of the means and the variances of the stretching stiffness and bending stiffness: $2 \times 4 \times 6 + 2 \times 3 \times 5 = 78$ learnable parameters.

The gravity, $\mathbf{F}_{gravity}$, is calculated on every face and evenly divided by its three vertices. Therefore, the gravity on the kth vertex is

$$\mathbf{F}^{(k)}_{gravity} = m^{(k)}\mathbf{g} = \sum_{j=0}^{n_j^{(k)}} \frac{1}{3}\rho^{(j)}A^{(j)}\mathbf{g} \tag{15}$$

where $\rho^{(j)}$ is the jth face's area density and $\mathbf{g} = [0.0, 0.0, -9.8]^\top m/s^2$. $n_j^{(k)}$ is number of the adjacent faces of vertex k (Fig. 12 Right). The handle force, $\mathbf{F}_{handle}$, is used to pin and support a cloth sample, i.e. simulating the inner support panel. To each vertex whose distance to the center of the cloth is smaller than 9cm, the handle force is computed as:

$$\mathbf{F}^{(k)}_{handles} = k_h\mathbf{I}_3(\mathbf{x}^{(k)} - \bar{\mathbf{x}}^{(k)}) \tag{16}$$

where $\bar{\mathbf{x}}^{(k)}$ denotes the kth vertex's anchor position where the vertex should be fixed and k_h is the handle stiffness.

7. Derivatives of the Simulator

Now we have a fully differentiable cloth simulator. We then compute the loss $\mathcal{L}$ that indicates the difference between the predicted and ground truth cloth states. The loss gradients with respect to the parameters $\frac{\partial \mathcal{L}}{\partial w}$ can help learn the right physics parameters via back-propagation. For simplicity, we use $\mathbf{Ay} = \mathbf{b}$ to represent Equation 7. The differential of $\mathbf{Ay} = \mathbf{b}$ is [6]:

$$\mathbf{A}\mathrm{d}\mathbf{y} = \mathrm{d}\mathbf{b} - \mathrm{d}\mathbf{A}\mathbf{y} \tag{17}$$

We can form the Jacobians of $\mathbf{y}$ with respect to $\mathbf{A}$ or $\mathbf{b}$ with Equation 17. For example, to compute the $\frac{\partial \mathbf{y}}{\partial \mathbf{A}}$, we need to set $\mathrm{d}\mathbf{A} = \mathbf{I}$ and $\mathrm{d}\mathbf{b} = \mathbf{0}$, then solve the equation and the result is $\frac{\partial \mathbf{y}}{\partial \mathbf{A}}$. As pointed out by Amos and Kolter [1], it is unnecessary to explicitly compute these Jacobians in back-propagation. We want to compute the product of the vector passed from back-propagation, $\frac{\partial \mathcal{L}}{\partial \mathbf{y}}$ and the Jacobians of $\mathbf{y}$,

(a) Training Sample (b) Initial Drape Silhouette (c) Trained Drape Silhouette (d) Cotton Blue Drape Silhouette (e) Paper Drape Silhouette

Figure 11. Through digitalization (*i.e.* training BDP), cloth drape silhouette (b and c) is optimized toward the training sample(a). In addition, (d) shows the simulated Cusick drape silhouette of a paper-like material.

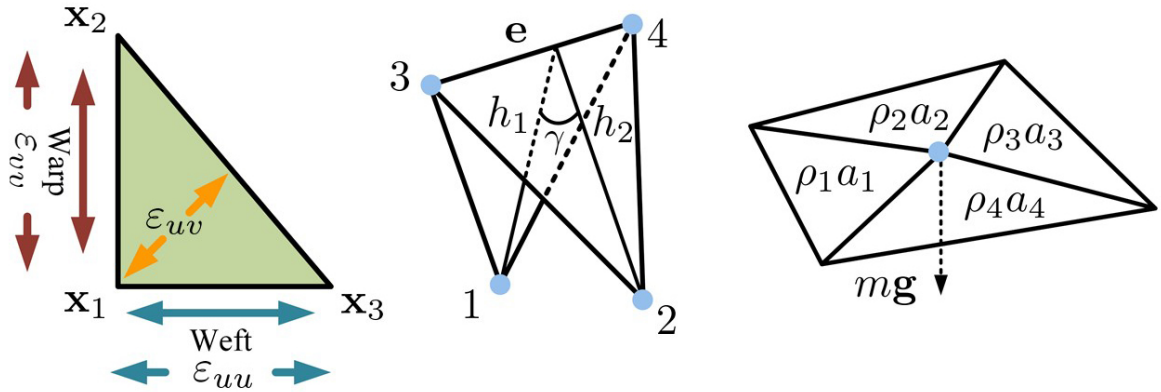


Figure 12. Left: cloth weft and warp directions (or course and wale directions in knitted cloths) and three strain directions in a triangle. Middle: bending force between two adjacent triangles. Right: triangle mass and gravity.

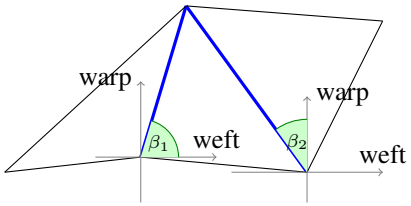


Figure 13. The bending bias angles, β_1 and β_2, of the two bending edges (blue).

i.e. $\frac{\partial \mathcal{L}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{A}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{b}}$. Assume $\mathbf{A} \in \mathbb{R}^{3\times3}$, $\mathbf{y} \in \mathbb{R}^3$, and $\mathbf{b} \in \mathbb{R}^3$, then

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{b}} = \left(\begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_1} & \frac{\partial \mathcal{L}}{\partial \mathbf{y}_2} & \frac{\partial \mathcal{L}}{\partial \mathbf{y}_3} \end{pmatrix} \begin{pmatrix} \frac{\partial \mathbf{y}_1}{\partial \mathbf{b}_1} & \frac{\partial \mathbf{y}_1}{\partial \mathbf{b}_2} & \frac{\partial \mathbf{y}_1}{\partial \mathbf{b}_3} \\ \frac{\partial \mathbf{y}_2}{\partial \mathbf{b}_1} & \frac{\partial \mathbf{y}_2}{\partial \mathbf{b}_2} & \frac{\partial \mathbf{y}_2}{\partial \mathbf{b}_3} \\ \frac{\partial \mathbf{y}_3}{\partial \mathbf{b}_1} & \frac{\partial \mathbf{y}_3}{\partial \mathbf{b}_2} & \frac{\partial \mathbf{y}_3}{\partial \mathbf{b}_3} \end{pmatrix} \right)^\top \tag{18}$$

As

$$\frac{\partial \mathbf{y}_1}{\partial \mathbf{b}_1} = \frac{\partial \left((\mathbf{A}^{-1})_{1,1}\mathbf{b}_1 + (\mathbf{A}^{-1})_{1,1}\mathbf{b}_2 + (\mathbf{A}^{-1})_{1,1}\mathbf{b}_3 \right)}{\partial \mathbf{b}_1} = \mathbf{A}^{-1}_{1,1}$$

and similarly for $\frac{\partial \mathbf{y}_i}{\partial \mathbf{b}_j}$, Equation 18 can be represented as:

$$\left(\begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_1} & \frac{\partial \mathcal{L}}{\partial \mathbf{y}_2} & \frac{\partial \mathcal{L}}{\partial \mathbf{y}_3} \end{pmatrix} \begin{pmatrix} (\mathbf{A}^{-1})_{1,1} & (\mathbf{A}^{-1})_{1,2} & (\mathbf{A}^{-1})_{1,3} \\ (\mathbf{A}^{-1})_{2,1} & (\mathbf{A}^{-1})_{2,2} & (\mathbf{A}^{-1})_{2,3} \\ (\mathbf{A}^{-1})_{3,1} & (\mathbf{A}^{-1})_{3,2} & (\mathbf{A}^{-1})_{3,3} \end{pmatrix} \right)^\top = (\mathbf{A}^{-1})^\top \frac{\partial \mathcal{L}}{\partial \mathbf{y}} \tag{19}$$

After computing $\frac{\partial \mathcal{L}}{\partial \mathbf{b}}$, we need to compute $\frac{\partial \mathcal{L}}{\partial \mathbf{A}}$. The $\mathbf{b}$ in Equation 17 can be set to 0 because it is irrelevant when computing $\frac{\partial \mathcal{L}}{\partial \mathbf{A}}$. Then we have

$$\mathbf{A}\mathrm{d}\mathbf{y} = -\mathrm{d}\mathbf{A}\mathbf{y} \tag{20}$$

The derivative of $\mathbf{y}$ with respect to $\mathbf{A}_{i,j}$, the entry in the ith row and jth column of the matrix $\mathbf{A}$, is

$$\frac{\partial \mathbf{y}}{\partial \mathbf{A}_{i,j}} = \mathbf{A}^{-1} \begin{pmatrix} 0 \\ -\mathbf{y}_j \\ 0 \end{pmatrix} \tag{21}$$

According to chain rule,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}_{i,j}} = \frac{\partial \mathcal{L}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{A}_{i,j}} = \frac{\partial \mathcal{L}}{\partial \mathbf{b}}^{\top} \mathbf{A}\mathbf{A}^{-1} \begin{pmatrix} \mathbf{0} \\ -\mathbf{y}_j \\ \mathbf{0} \end{pmatrix}$$
$$= -\left(\frac{\partial \mathcal{L}}{\partial \mathbf{b}}\right)_i \mathbf{y}_j \tag{22}$$

The more general form is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = -\frac{\partial \mathcal{L}}{\partial \mathbf{b}} \mathbf{y}^{\top} \tag{23}$$

References

[1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017. 9

[2] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998. 7

[3] Javier Bonet and Richard D Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008. 8

[4] Xudong Feng, Wenchao Huang, Weiwei Xu, and Huamin Wang. Learning-based bending stiffness parameter estimation by a drape tester. *ACM Transactions on Graphics (TOG)*, 41 (6):1–16, 2022. 1

[5] Junbang Liang, Ming Lin, and Vladlen Koltun. Differentiable cloth simulation for inverse problems. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[6] Jan R Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019. 9

[7] XQ Peng and Jian Cao. A continuum mechanics-based non-orthogonal constitutive model for woven composite fabrics. *Composites part A: Applied Science and manufacturing*, 36 (6):859–874, 2005. 8

[8] Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics*, 28 (4):Article–No, 2009. 7

[9] Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011. 8