Learning-Based Bending Stiffness Parameter Estimation by a Drape Tester

XUDONG FENG, State Key Laboratory of CAD&CG, Zhejiang University, China and Style3D Research, China
WENCHAO HUANG, Style3D Research, China
WEIWEI XU, State Key Laboratory of CAD&CG, Zhejiang University, China
HUAMIN WANG, Style3D Research, China

(a) Fabric #1 (b) Fabric #10 (c) Fabric #17 (d) Fabric #24

Fig. 1. A long dress example simulated by using four fabrics with different bending stiffness properties. Based on the Cusick drape test method, we develop a novel learning-based system to estimate bending stiffness parameters from multi-view depth images defining the drape of a real-world fabric specimen. Compared with traditional cantilever tests, our system is not only more reliable and effective, but also easier to use. On average, the use of our system reduces the parameter acquisition time from 15 minutes to under 3 minutes, and helps our simulator achieve higher fidelity as shown in Section 7.

Real-world fabrics often possess complicated nonlinear, anisotropic bending stiffness properties. Measuring the physical parameters of such properties for physics-based simulation is difficult yet unnecessary, due to the persistent existence of numerical errors in simulation technology. In this work, we propose to adopt a simulation-in-the-loop strategy: instead of measuring the physical parameters, we estimate the simulation parameters to minimize the discrepancy between reality and simulation. This strategy offers good flexibility in test setups, but the associated optimization problem is computationally expensive to solve by numerical methods. Our solution is to train a regression-based neural network for inferring bending stiffness parameters, directly from drape features captured in the real world. Specifically, we choose the Cusick drape test method and treat multiple-view depth images as the feature vector. To effectively and efficiently train our network, we develop a highly expressive and physically validated bending stiffness model, and we use the traditional cantilever test to collect the parameters of this model for 618 real-world fabrics. Given the whole parameter data set, we then construct a parameter subspace, generate new samples within the subspace, and finally simulate and augment synthetic data for training purposes. The experiment shows that our trained system can replace cantilever tests for quick, reliable and effective estimation of simulation-ready parameters. Thanks to the use of the system, our simulator can now faithfully simulate bending effects comparable to those in the real world.

CCS Concepts: • **Computing methodologies → Physical simulation**.

Additional Key Words and Phrases: cloth simulation, bending stiffness, parameter estimation, deep neural networks, material property acquisition

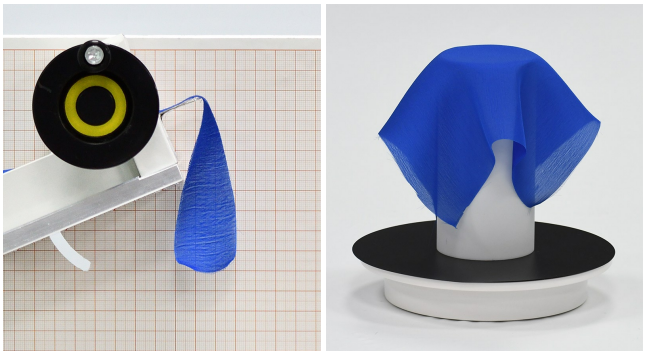
ACM Reference Format:
Xudong Feng, Wenchao Huang, Weiwei Xu, and Huamin Wang. 2022. Learning-Based Bending Stiffness Parameter Estimation by a Drape Tester. *ACM Trans. Graph.* 41, 6, Article 1 (December 2022), 16 pages. https://doi.org/10.1145/3550454.3555464

Authors' addresses: Xudong Feng, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China, Style3D Research, Hangzhou, China, xudongfeng@zju.edu.cn; Wenchao Huang, Style3D Research, Hangzhou, China, physhuangwenchao@gmail.com; Weiwei Xu, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China, xww@cad.zju.edu.cn; Huamin Wang, Style3D Research, Hangzhou, China, wanghmin@gmail.com.

0730-0301/2022/12-ART1 $15.00
https://doi.org/10.1145/3550454.3555464

1 INTRODUCTION

Performance and fidelity are the two key criteria judging the success of physics-based cloth simulation engines. While graphics researchers [Bouaziz et al. 2014; Wang and Yang 2016; Wu et al. 2020] have accomplished substantial success on improving simulation performance, their recent progress on simulation fidelity is rather



(a) A curly fabric in the cantilever test (b) A curly fabric in the drape test

Fig. 2. A curly fabric tested by two methods. While bending stiffness of a curly fabric becomes difficult to estimate in the cantilever test, it is still doable in the drape test shown in (b).

limited. This reality is particularly harsh to digital fashion designers and developers, as they need high-fidelity simulators to create virtual garments similar, if not identical, to real garments. While many factors can affect the simulation fidelity, planar and bending stiffness properties are the two undeniably critical ones. The planar stiffness property is more important to elastic fabrics, which are typically used to make underwear and sportswear. In contrast, the bending stiffness property is important to almost every fabric, due to its high influence on wrinkle details. Since real-world bending stiffness is nonlinear, anisotropic and varies dramatically across different fabrics, how to estimate it with reliability and effectiveness is a major challenge.

Over the past few decades, fabric engineers and scientists have developed a variety of standard test methods [ASTM 2016, 2018; ISO 2008] relevant to bending stiffness, including the cantilever method, the heart-loop method, the drape method and the hanging method. Compared with other methods, the cantilever method is probably the most popular and intuitive one. Assuming that bending behaviors in different material directions are isolated, a cantilever tester [Kato Tech 2022; Taber Industries 2022] uses an overhang to evaluate how much a fabric strip bends under its own weight as shown in Fig. 2a. In computer graphics [CLO 2022; Romero et al. 2021], the cantilever method is also the leading method used in research and development. Unfortunately, the cantilever method suffers from several severe shortcomings.

- *Time cost.* Our experiment shows it takes at least 15 minutes for an experienced user to test one fabric by a cantilever tester. This includes the time spent on preparing strip specimens and the time spent on the actual test. For a fabric manufacturer in need of digitizing thousands of fabrics in stock, such a time cost is unaffordable.
- *Curly effects.* As pointed out by Wang et al. [2011], many fabric strips cannot be properly evaluated by a cantilever tester, due to curly effects shown in Fig. 2a. Fundamentally, it is physically incorrect to model bending stiffness separately in multiple material directions.
- *Simulation error.* Many cantilever-based parameter estimation approaches measure bending stiffness as an intrinsic property of a fabric. However, a physics-based simulator has its own error. Therefore, even if the measurement is perfect, the simulated behavior can still be far from reality.

In this paper, we switch our focus to an alternative: the Cusick drape method [1965]. Compared with other methods, the drape method can more naturally reveal wrinkles and folds similar to those needed in simulation, by draping a fabric specimen onto a cylindrical platform as Fig. 2b shows. Square or circular fabric specimens are also more accessible from fabric swatch books, frequently used by designers to find desired fabrics. Traditionally, the drape method is not considered to be suitable for parameter estimation, but for quantifying bending stiffness by various indicators [Carrera-Gallissa et al. 2017]. We can overcome this issue by adopting a *simulation-in-the-loop* parameter estimation strategy:

> *given the drape of a real-world fabric evaluated by the drape method, we would like to find the optimal bending parameters for a specific simulator to reproduce the identical drape in simulation.*

Solving the optimization problem for simulation-in-the-loop parameter estimation by numerical methods [Clyde et al. 2017; Wang et al. 2011] is not only difficult but also time consuming, due to the complex relationship between parameters and shapes. Inspired by recent research on learning-based fabric parameter estimation [Ju and Choi 2020; Rasheed et al. 2020, 2021; Yang et al. 2017], we propose to solve this problem using deep neural networks. In particular, we argue that the quasistatic shapes of real-world fabrics in drape tests are much less diversified than those in other test environments. Therefore we can build a reliable and effective learning-based parameter estimation system, while still keep the size of the needed data set tractable. To achieve this goal, we make a series of technical contributions.

- *Nonlinear, anisotropic bending stiffness.* We study how to generalize popular bending models into nonlinear, anisotropic ones. This is crucial to the existence of suitable parameters for a simulator to reproduce bending behaviors. For comparison and data collection purposes, we also present cantilever-based techniques for estimating parameters of these models.
- *Synthetic data generation.* We apply a cantilever tester to obtain rough parameters of hundreds of real-world fabrics and use them to construct a subspace. Given the subspace and the simulator, we generate a large synthetic data set, with data augmentation addressing local minima, unknown fabric orientation and sensing errors.
- *Drape tester and neural network.* We develop a novel drape tester and we capture multi-view depth images as the feature vector. Using the regression neural network trained by the synthetic data set, we show how to quickly infer bending parameters of a real fabric evaluated by our tester.

In addition, we provide a comprehensive study on the cost, the reliability and the effectiveness of our system. The experiment shows our system is reliable, easy to use, and more importantly, effective in improving the fidelity of simulated bending behaviors.

In a user study, designers and artists concur that our system provides more plausible parameters than a cantilever tester for most of the fabrics. Thanks to the system, the estimation process of a fabric also becomes significantly shorter, typically under three minutes from preparing a specimen to obtaining the parameters. (Supplemental code and datasets for this work is downloadable at: https://github.com/DrapeTester/ClothDrapeTester.)

2 PREVIOUS WORK

Physics-based cloth simulation. Due to its key role in graphics applications, physics-based cloth simulation has been an important research topic since the seminal work by Baraff and Witkin [1998]. While early cloth simulators [Bridson et al. 2002, 2003; Selle et al. 2009] often choose to integrate cloth dynamics explicitly over time, recent simulators [Liu et al. 2013; Narain et al. 2012; Volino et al. 2009] typically adopt implicit time integration for numerical stability with large time steps. In general, cloth simulators treat planar and bending deformations as two independent modes. Popular planar stiffness models are the mass-spring model [Choi and Ko 2002] and continuum-based models [Müller et al. 2005; Volino et al. 2009], and popular bending stiffness models are the dihedral angle model [Bridson et al. 2003; Tamstorf and Grinspun 2013] and the quadratic bending model [Bergou et al. 2006]. In recent years, researchers have made substantial progress in speeding up cloth simulation and their achievements include: position-based dynamics [Müller et al. 2005], projective dynamics [Bouaziz et al. 2014] and real-time dynamic solvers on GPUs [Fratarcangeli et al. 2016; Wang 2015; Wang and Yang 2016]. Given their success, we believe it is now the time to push the limit of the simulation fidelity further.

We would like to emphasize that there are many other cloth simulation topics, such as yarn-based cloth simulation [Kaldor et al. 2008, 2010] and contact handling of cloth [Bridson et al. 2002; Tang et al. 2018]. These topics are less relevant to this work, but how to acquire simulation parameters involved in yarns and contacts is also an important problem worthy of investigation.

Physical parameter estimation. Early graphics research on parameter estimation of fabrics [Bhat et al. 2003; Kunitomo et al. 2010] often tried to acquire the parameters of multiple physical properties simultaneously from unconstrained cloth movements. This turned out to be a highly challenging problem, due to mutual influence among the properties. Being aware of this, Wang et al. [2011] proposed to estimate planar and bending stiffness by two separate tests. Miguel et al. [2012] developed a more automatic device to acquire fabric stiffness parameters. Using the same device, they [2013] modeled and captured fabric internal friction for hysteresis later. Chen et al. [2013] developed another device to measure friction between fabrics and solids. Ly et al. [2018] studied inverse shape estimation of cloth and shells with contact and friction. Recently, Clyde et al. [2017] investigated hyperelastic models for planar stiffness of woven fabrics and its parameter estimation by tensile tests. They tried to approximate bending stiffness under the Kirchhoff-Love hypotheses as well, but the resulting fidelity is limited.

Physical parameter estimation of deformable bodies, especially human tissues, is also an important problem in computer graphics [Bickel et al. 2009; Pai et al. 2001, 2018; Wang et al. 2015]. While the stiffness properties of deformable bodies share many similarities with those of fabrics, such as nonlinearity and anisotropy, their estimation problems have their own features and challenges.

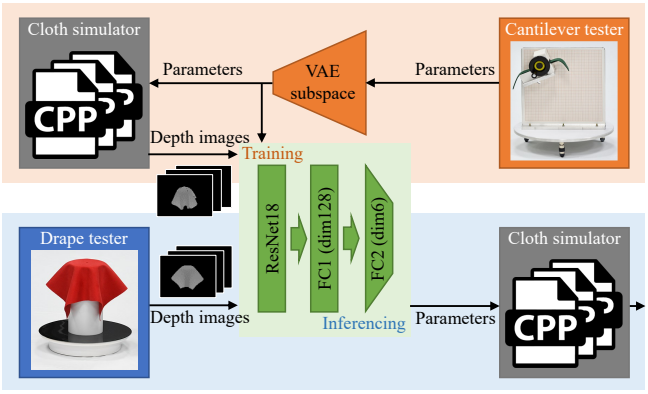


Fig. 3. The pipeline. Our system uses a simulator to generate multi-view depth images of draped specimens as synthetic training data, given the parameters sampled in a VAE subspace. The trained network can then make bending stiffness inferences from the depth images of a real fabric specimen captured by a drape tester. The estimated parameters allow our simulator to reproduce nonlinear, anisotropic bending effects in high fidelity.

Instead of solving parameter estimation as a numerical optimization problem, researchers have also explored the use of machine learning to infer physical parameters, such as stiffness of rock [Shahriari et al. 2010], fluid viscosity [Changdar et al. 2021; Mohan M S and Menon 2021] and surface reflectiveness [Gao et al. 2019; Kang et al. 2021]. To estimate fabric stiffness from controlled movements in hanging tests, Bouman et al. [2013] and Davis et al. [2017] used linear regression without separating planar stiffness from bending stiffness. To estimate bending stiffness from less controlled experiments, Yang et al. [2017] developed a learning-based system without considering nonlinearity nor anisotropy. Recently, Rui et al. [2020] adopted learning techniques to predict fabric physical parameters solely from yarn composition and finishing features. Rasheed et al. [2020; 2021] investigated learning-based parameter estimation of both fabric material and friction from controlled movements. Similar to our work, Ju and Choi [2020] applied deep neural networks to estimate bending stiffness parameters of real-world fabrics by drape tests. Their method ignores nonlinearity and local minima, and it defines the feature vector as the boundary curve of a specimen, which is difficult to acquire in practice and insufficient to cover all of the shape details.

3 OVERVIEW

Fig. 3 shows an overview of our learning-based bending parameter estimation system. The purpose of this system is to provide suitable bending stiffness parameters to a physics-based cloth simulator, so that it can simulate high-fidelity bending behaviors of a real-world fabric. To do so, we capture the drape shape of a fabric specimen by multi-view depth images using a drape tester, and treat them as the feature vector for our deep neural network to make inferences.

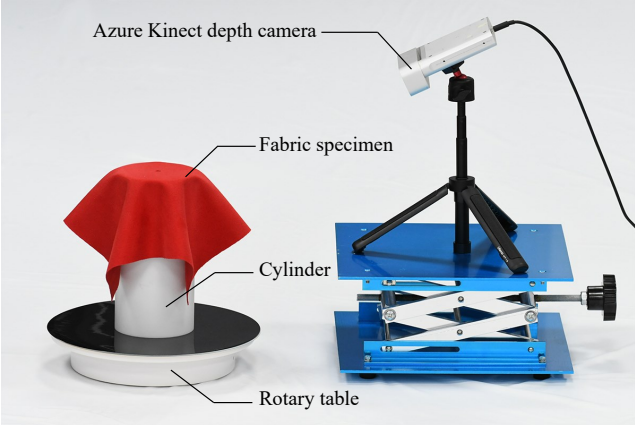


Fig. 4. A drape tester. Our drape tester includes a cylinder placed on a motorized rotary table and an Azure Kinect depth camera. Using the table and the camera, we capture depth images of the draped fabric specimen in four views, as the feature vector encoding bending stiffness.

The key question is how to train such a deep neural network. Building a large training data set from the real world would be too time consuming and fail to consider the intrinsic errors involved in acquisition and simulation. Instead we use our physics-based cloth simulator to generate a large number of multi-view depth images as synthetic training data, given a set of stiffness parameter samples. The follow-up question is: how to generate such parameter samples? To model bending stiffness of a real-world fabric, we use nonlinear anisotropic models with up to six parameters, which form a full space too large to be directly sampled. Therefore, we train a parameter subspace using the parameters roughly measured by a cantilever tester, and then generate samples in that subspace. Our experiment validates the effectiveness of the system and the accuracy of our neural network.

3.1 A Drape Tester

Our drape tester is shown in Fig. 4. The key component of this tester is a cylinder with a diameter of 100mm, placed on a motorized rotary table. This table can rotate steadily with a maximum speed of $\pi/10$ radians per second with remote control. To test bending stiffness of a real-world fabric, we prepare a 300mm×300mm fabric specimen in alignment with the two basic weaving or knitting directions, known as *warp* and *weft*, and drape it on top of the cylinder. We require that the specimen center matches the cylinder center, but the specimen orientation can be arbitrary. After we drape the specimen, we capture four depth images at the resolution of 240×180 with an equally rotated gap of $\pi/2$ by an Azure Kinect depth camera [Microsoft 2022]. These four depth images form the feature vector describing the bending properties of the fabric. We assume that the table and the camera have been set to fixed positions and orientations for a clear view of the drape. We calibrate the camera ahead of time.

4 SIMULATOR AND STIFFNESS MODELS

To begin with, we would like to discuss our simulator and its stiffness models for data synthesis and result production. Our simulator is an in-house GPU-based one, which runs a series of Newton-Raphson iterations to solve dynamic simulation or quasistatic equilibrium problems. Our simulator accepts a variety of stiffness models and our parameter estimation system works with most of them. Having said that, we still prefer the actual models to be plausible and expressive enough so that our simulator can reproduce as many real-world effects as possible given suitable parameters.

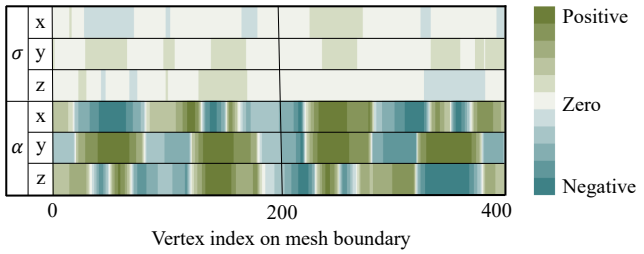
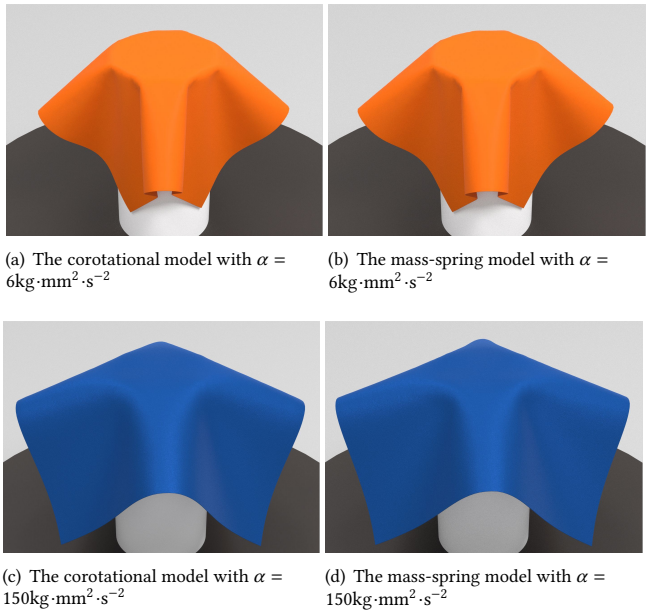


Fig. 5. The correlations between the in-plane and bending stiffness parameters (σ and α) and the simulated mesh vertex positions in 3D. To calculate such correlations, we run the simulation multiple times, with 50 σ samples uniformly chosen between $100\text{kg}\cdot\text{s}^{-2}$ and $1,000\text{kg}\cdot\text{s}^{-2}$, and 50 α samples logarithmically chosen between $100\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$ and $10\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$. Compared with the bending stiffness parameter, the in-plane stiffness parameter has much less influence on the simulated drape mesh.



(a) The corotational model with $\alpha = 6\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$

(b) The mass-spring model with $\alpha = 6\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$

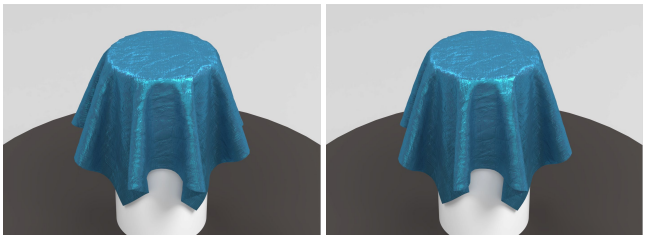
(c) The corotational model with $\alpha = 150\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$

(d) The mass-spring model with $\alpha = 150\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$

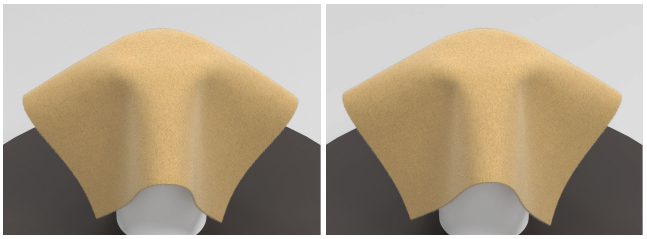
Fig. 6. The drapes simulated with different in-plane stiffness models and different bending stiffness parameter values. This figure shows that the influence of in-plane models on the drape shape is limited.

4.1 In-Plane Stiffness

To develop our system, we make a crucial assumption that a fabric specimen barely stretches under its own weight during the drape test. This assumption suggests that in-plane stiffness models and their parameter values do not matter to drape simulation. We justify this



(a) $\sigma = 200\text{kg}\cdot\text{s}^{-2}$, $\alpha = 600\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$ (b) $\sigma = 2,000\text{kg}\cdot\text{s}^{-2}$, $\alpha = 600\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$



(c) $\sigma = 200\text{kg}\cdot\text{s}^{-2}$, $\alpha = 60\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$ (d) $\sigma = 2,000\text{kg}\cdot\text{s}^{-2}$, $\alpha = 60\text{kg}\cdot\text{mm}^2\cdot\text{s}^{-2}$

Fig. 7. The drapes simulated with different values of in-plane and bending stiffness parameters. Unlike the bending parameter α, the in-plane parameter σ has significantly smaller influence on the drape shape.

assumption in Fig. 5, which reveals that the correlation between the in-plane stiffness parameter σ and the simulated drape mesh is weak. Meanwhile, Fig. 6 and 7 demonstrate that the choice of in-plane stiffness models and the magnitude of σ hardly affect the simulated drape mesh visually, as long as σ is sufficiently large. Based on this assumption, we choose to use the corotational FEM model with a uniform isotropic in-plane stiffness parameter $\sigma = 200\text{kg}\cdot\text{s}^{-2}$ by default.

We note that many simulators suffer from the locking issue, i.e., large in-plane stiffness improperly stiffening bending behaviors, due to a lack of degrees of freedom. To address this issue in our simulator, we avoid very large in-plane stiffness parameters, and allow cloth to deform freely within a ratio of [99%, 101%] by dynamically adjusting the reference state during simulation. These practices are effective as Fig. 9 shows. To further lessen the locking issue, our simulator can adopt higher-order or non-conformal elements [English and Bridson 2008] as well, which are not necessary yet.

4.2 Uniaxial Bending Stiffness

Let us consider bending resistance in a single material direction first. Fig. 8 shows a bending element made of two adjacent triangles. Suppose that the element is planar in the reference state and the only bending mode is to rotate around the hinged edge. Based on this element, we will discuss two popular bending stiffness models: the dihedral angle model [Bridson et al. 2003; Grinspun et al. 2003; Tamstorf and Grinspun 2013] and the quadratic model [Bergou et al. 2006], and their parameter estimation by a cantilever tester.

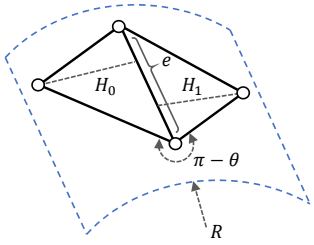


Fig. 8. A bending element made of two adjacent triangles.

(a) Before fixing the locking issue (b) After fixing the locking issue

Fig. 9. The simulated drapes before and after fixing the locking issue. When the in-plane stiffness parameter σ is large, i.e., $\sigma = 5,000\text{kg}\cdot\text{s}^{-2}$, a simulator can produce locking artifacts shown in (a). We must eliminate this issue before we can apply the simulator in bending parameter estimation.

4.2.1 The dihedral angle model. Let $\tau(\kappa, e)$ be the torque as a function of the unsigned curvature κ and the hinged edge length e, with $\tau(0, e) = 0$ and $\tau(\kappa, 0) = 0$. The energy of a bending element is:

$$E(\theta) = \int_0^\theta \tau(\kappa(\xi), e) d\xi, \tag{1}$$

where θ is the dihedral angle. In our system, we assume that $\tau(\kappa, e)$ is in the form of $(\alpha\kappa + \beta\kappa^2)e$, as a linear function of e and a quadratic function of κ. Here α and β are the linear and quadratic bending moduli. From Eq. 1, we calculate the force at vertex i as:

$$\mathbf{f}_i = -\nabla_i E = -\tau(\kappa, e)\nabla_i\theta(\mathbf{x}) = -(\alpha\kappa + \beta\kappa^2)e\nabla_i\theta(\mathbf{x}). \tag{2}$$

To utilize Eq. 2, we need to calculate κ. Similar to [Narain et al. 2012], we assume that cloth deformation is nearly isometric, the element is small and $\theta \approx 0$. Therefore, we estimate κ from the radius R of a cylinder approximating the element as Fig. 8 shows:

$$\kappa = \frac{1}{R} = \frac{2\sin\theta}{\left(H_0^2 + H_1^2 - 2H_0H_1\cos(\pi - \theta)\right)^{\frac{1}{2}}} \approx \frac{2\theta}{H_0 + H_1}, \tag{3}$$

in which H_0 and H_1 are two constant reference triangle heights. Eq. 3 shows that κ is a linear function of θ.

We treat this nonlinear dihedral angle model as our default choice in the system. Its linear version (when $\beta = 0$) is mathematically equivalent to the one used by ARCSim [Narain et al. 2012] or the discrete shell model [Grinspun et al. 2003], except for a constant factor of 4 or 1/3. Therefore, we can convert $[\alpha \quad \beta]$ under our model to $[4\alpha \quad 16\beta]$ for ARCSim, or $[\alpha/3 \quad \beta/9]$ under the discrete shell model, to achieve identical simulation outcomes. The reason we still stick to our model is because it is the most plausible one, according to the cantilever-based physical validation experiment [Romero et al. 2021]. In this experiment, we simulate fabric strips of varying lengths and plot out simulated strips based on the relationship between the aspect ratio of the deflection curve and the dimensionless gravito-bending parameter $\Gamma = \rho g L^3/\alpha$, where ρ is the fabric density, g is the gravity acceleration and L is the strip length. Fig. 10

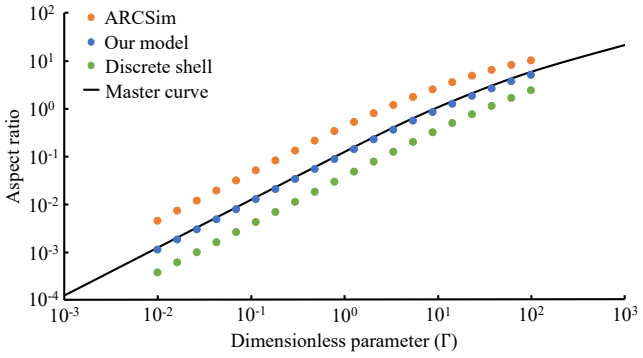
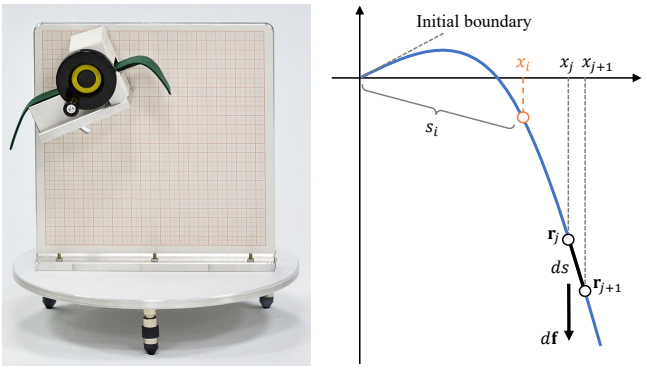


Fig. 10. Physical validation of three dihedral angle bending models. According to the cantilever-based experiment [Romero et al. 2021], we simulate multiple fabric strips and illustrate them as dots, based on the relationship between the aspect ratio of the deflection curve and the dimensionless parameter Γ. This figure shows that the strips simulated by using our model are the most consistent ones with the ground-truth master curve.



(a) Our cantilever tester (b) A deflection curve

Fig. 11. Our cantilever tester and the deflection curve of a strip specimen. We build our cantilever tester shown in (a) to evaluate uniaxial bending stiffness from the deflection curve of a fabric strip specimen under its own weight, as shown in (b). To evaluate anisotropic bending stiffness, we need to test multiple strip specimens.

shows that a simulator using our model generates the most consistent strips with the master curve[1], i.e., the numerical solution to the planar elastica [Romero et al. 2021].

4.2.2 The quadratic model. The quadratic model [Bergou et al. 2006] is another popular choice for modeling bending stiffness, under the nearly isometric deformation assumption. Instead of using the dihedral angle, it defines the energy of a bending element as a quadratic function of the four vertex positions $\mathbf{x} \in \mathbb{R}^{12}$:

$$E(\mathbf{x}) = \frac{3\gamma}{(H_0 + H_1)e} \|\mathbf{K}\mathbf{x}\|^2, \tag{4}$$

in which $\mathbf{K} \in \mathbb{R}^{3\times 12}$ is a constant cotangent weight matrix and γ is its bending stiffness parameter. According to Eq. 4, the bending force is a linear function of $\mathbf{x}$.

4.3 Cantilever-Based Parameter Estimation

For comparison and data collection purposes, we would like to estimate the parameters of a bending stiffness model by a cantilever tester as well. As Fig. 11a shows, our tester is made of an adjustable ramp, a grid board and a DSLR camera with a telescopic lens shooting from a distance. The ramp is adjustable to reach different slopes. The specimen of a real fabric is a 200mm×30mm strip. Let the deflection curve of one strip specimen be shown in Fig. 11b. We manually select four control nodes over it and fit the nodes by a cubic Bezier curve. We then uniformly sample the curve along the x-axis to obtain N sample points $\{\mathbf{r}_1, ..., \mathbf{r}_N\}$. In our system, $N = 256$. Given these sample points, we describe two parameter estimation methods for bending models next.

4.3.1 Parameter estimation by regression. To estimate the bending moduli under the dihedral angle model, we propose to solve a regression problem. By definition, the torque magnitude at sample $\mathbf{r}_i$ can be calculated as:

$$\tau_i = E\left\| \int_{s_i}^{s_N} (\mathbf{r}(s) - \mathbf{r}_i) \times d\mathbf{f}(s) \right\| = \rho g E \int_{s_i}^{s_N} (x(s) - x_i) ds, \tag{5}$$

in which ρ is the fabric density, g is the gravity acceleration, E is the strip width, s is the arc length variable and s_i and s_N are the arc lengths at $\mathbf{r}_i$ and $\mathbf{r}_N$. By applying trapezoidal rule to Eq. 5, we obtain τ_i at every sample:

$$\tau_i = \rho g E \sum_{j=i}^{N-1} \frac{x_j + x_{j+1} - 2x_i}{2} \|\mathbf{r}_{j+1} - \mathbf{r}_j\|. \tag{6}$$

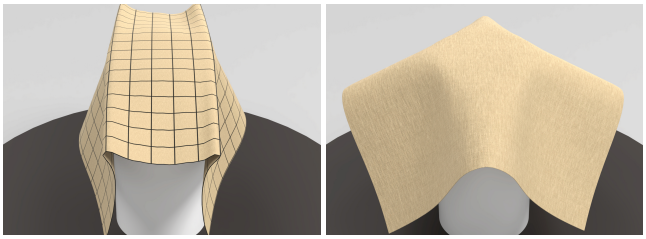
Meanwhile, the model in Subsection 4.2.1 specifies $\tau = (\alpha\kappa + \beta\kappa^2)E$, where α and β are the two unknown bending moduli. Given the curvature κ_i and the torque τ_i estimated at every sample, we solve a quadratic regression problem to obtain α and β.

4.3.2 Parameter estimation by classification. The regression method is unsuitable for the models not using bending moduli as their parameters, such as the quadratic model. In that case, we propose an alternative method based on classification. Specifically, we define 1,000 categories per bending parameter and 100 curve lengths (from 1mm to 100mm), and then run simulation to generate corresponding deflection curves. In total, we obtain a data set containing 100,000 parameter-deflection pairs for a single-parameter bending model. During parameter estimation, we then measure the curve length, find the most similar curve to the real one by nearest neighbor search, and finally output its associated parameter.

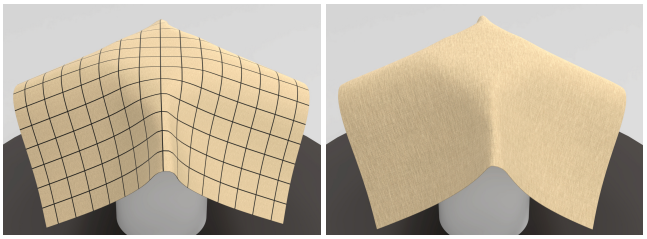
4.4 Anisotropic Bending Stiffness

Given the uniaxial models for a single bending element, including both the dihedral angle model and the quadratic model, we would like to generalize them into anisotropic ones next. A common way is to simply assign different stiffness parameters to different bending elements, based on hinged edge orientations in the reference state [Bergou et al. 2006; Wang et al. 2011]. But such an edge-based

[1]Please check the supplemental MATLAB code to replicate this experiment. We do not evaluate the early model based on the use of $\sin(\theta/2)$ [Bridson et al. 2003], since it should be much less consistent with the master curve as $|\theta|$ increases.

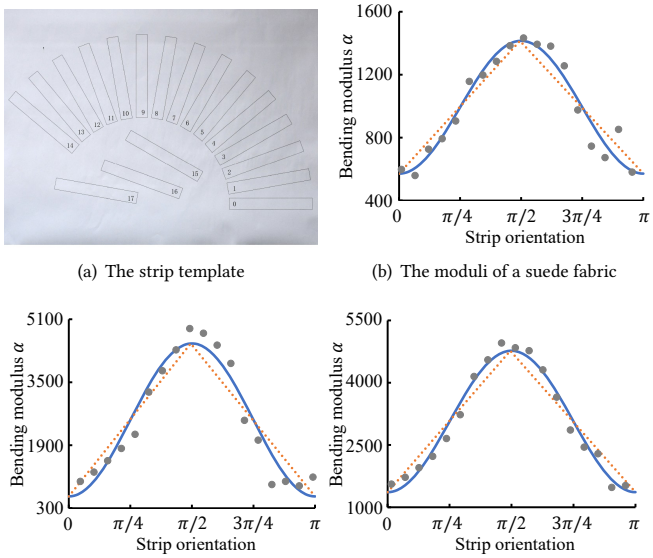


(a) Grid-structured mesh with edge-based anisotropic bending stiffness (b) Unstructured mesh with edge-based anisotropic bending stiffness



(c) Grid-structured mesh with curvature-based anisotropic bending stiffness (d) Unstructured mesh with curvature-based anisotropic bending stiffness

Fig. 12. The simulated drapes with anisotropic bending stiffness implemented by two approaches. While the edge-based approach causes simulation results to be mesh-dependent as shown in (a) and (b), the curvature-based approach produces consistent results as shown in (c) and (d), even when one mesh is structured while the other one is not.



(a) The strip template (b) The moduli of a suede fabric

(c) The moduli of medium-weight denim (d) The moduli of a coated cotton fabric

Fig. 13. An anisotropic bending experiment. With the help of a paper template in (a), we prepare 18 strip specimens for each fabric and measure their bending moduli (in gray) by a cantilever tester (in Subsection 4.3). The results from (b) to (d) indicate that our method (in blue) approximates the moduli better than linear interpolation (in orange) does.

approach fails to model bending anisotropy correctly, since the orientation of an individual edge may not match the local bending direction from a geometric perspective. In simulation, such an error can cause bending behaviors to be mesh-dependent as Fig. 12a and 12b show.

Our solution is to calculate the principal curvatures and their directions at all of the vertices [Meyer et al. 2003], and estimate the bending direction of a dihedral element as the mean of the greater principal directions at the two edge vertices. Let $\mathbf{k}^{\text{warp}} = [\alpha^{\text{warp}} \quad \beta^{\text{warp}}]$ and $\mathbf{k}^{\text{weft}} = [\alpha^{\text{weft}} \quad \beta^{\text{weft}}]$, or $\mathbf{k}^{\text{warp}} = [\gamma^{\text{warp}}]$ and $\mathbf{k}^{\text{weft}} = [\gamma^{\text{weft}}]$ under the quadratic model, be the parameter vectors in warp and weft directions, respectively. Similar to the curvatures, the parameter vector $\mathbf{k}(\varphi)$ in an arbitrary bending direction φ can be approximated by:

$$\mathbf{k}(\varphi) = \cos^2\varphi\ \mathbf{k}^{\text{warp}} + \sin^2\varphi\ \mathbf{k}^{\text{weft}}, \tag{7}$$

as demonstrated by our strip experiment shown in Fig. 13. To make our anisotropic model even more expressive, we construct our model by bending parameters in three sample directions: $\mathbf{k}^{\text{warp}} \equiv \mathbf{k}(0)$, $\mathbf{k}^{\text{bias}} \equiv \mathbf{k}(\pi/4)$ and $\mathbf{k}^{\text{weft}} \equiv \mathbf{k}(\pi/2)$. Let φ be a material direction between any two sample directions φ_0 and φ_1. We calculate its parameter vector as:

$$\mathbf{k}(\varphi) = \begin{bmatrix} \cos^2\varphi \\ \sin^2\varphi \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \cos^2\varphi_0 & \sin^2\varphi_0 \\ \cos^2\varphi_1 & \sin^2\varphi_1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{k}^{\varphi_0} \\ \mathbf{k}^{\varphi_1} \end{bmatrix}, \tag{8}$$

Intuitively, Eq. 8 estimates the bending parameters in warp and weft directions first, and uses them to compute $\mathbf{k}(\varphi)$. Similar to [Wang et al. 2011], we assume that bending behaviors are symmetrical about warp and weft directions, i.e., $\mathbf{k}(\varphi) \equiv \mathbf{k}(\pi + \varphi) \equiv \mathbf{k}(\pi - \varphi) \equiv \mathbf{k}(-\varphi)$, so we can calculate the parameter vector for $\varphi \notin [0, \pi/2]$ as well. The joint parameter vector describing nonlinear, anisotropic bending stiffness of the whole fabric is six-dimensional: $\mathbf{k} = [\mathbf{k}^{\text{warp}} \quad \mathbf{k}^{\text{bias}} \quad \mathbf{k}^{\text{weft}}]$.

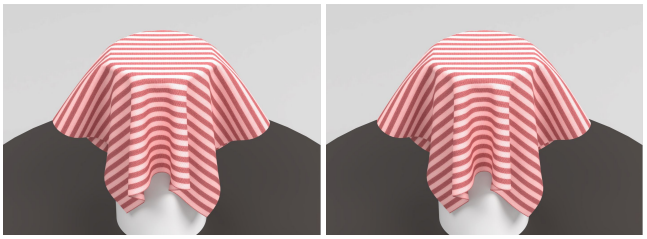
Fig. 12c and 12d show that our curvature-based approach effectively lessens the mesh-dependency issue. It does have a side effect: when cloth is nearly flat, the computed principal curvature directions are unreliable, leading to spatial oscillations in the bending parameters. These oscillations are especially noticeable if the material is extremely anisotropic. Fortunately, it is uncommon for real fabrics to possess such properties, as shown in Subsection 5.1.3.

5 SYNTHETIC DATA GENERATION

Next we study the generation of synthetic depth image data by simulation, for training our neural network later. The whole data generation process can be divided into two steps: parameter subspace construction (in Subsection 5.1), and data synthesis by using the parameters sampled in the subspace (in Subsection 5.2).

5.1 Parameter Subspace Construction

The very first challenge we face in data generation is a large parameter space. Even if we choose linear models with three parameters only and sample each parameter by 32 values, we would still end up with 32K data samples needed for simulation. Such a large data set would be more unaffordable to simulate, given the nonlinearity of the actual model and the need for data augmentation. Fortunately,



(a) Simulation with $\rho = 50\text{g}\cdot\text{m}^{-2}$ (b) Simulation with $\rho = 500\text{g}\cdot\text{m}^{-2}$

Fig. 14. The simulated drapes with different density values but the same $\mathbf{k}/\rho$. This example verifies that the simulator produces the identical outcome in the drape test, as long as the ratio of $\mathbf{k}$ to ρ stays the same.

most of the real fabrics exhibit similar bending behaviors. This implies that bending parameters reside in a small subspace.

The main problem is how to obtain the parameters of real fabrics needed for constructing the subspace, when our parameter estimation system is not ready in the first place. To address this problem, we resort to the cantilever method described in Subsection 4.3, which is not expected to be sufficiently reliable but can still provide rough estimations for subspace construction. In Subsection 5.1.2, we show how to reduce the amount of data needed for subspace construction, by eliminating the influence of fabric density. In Subsection 5.1.3, we evaluate different subspace models and their constructions.

5.1.1 Cantilever-based training data acquisition. We use the cantilever method in Subsection 4.3 to estimate the bending parameters of a real fabric in the warp, bias and weft material directions. The result is a six-dimensional parameter vector if the bending model is nonlinear, or a three-dimensional parameter vector if the model is linear. On average, it takes 15 man-minutes to measure one fabric, which includes the time spent on specimen preparation, cantilever tests and parameter estimation under various bending models. Overall, our data set contains the parameters of 618 real-world fabrics commonly used for garments. The whole measurement process takes more than 150 man-hours.

We would like to emphasize that when it estimates bending parameters for a specific fabric, the cantilever method suffers from many issues outlined in Section 1, including curly effects, not always being *simulation-in-the-loop* and hysteresis. But since the sole use of the cantilever method is to reduce the burden of data generation in a large parameter space, its effectiveness should have little impact on our system, as long as the overall estimation provides a sufficient coverage of the parameter subspace of real fabrics. As shown later in Subsection 7.2, our system provides more effective parameter estimation than the cantilever method.

5.1.2 Parameter normalization. In both cantilever and drape tests, the shape of a fabric specimen depends on not only bending parameters, but also its self weight. The self weight is fundamentally determined by the density ρ, which varies from fabric to fabric. If we define $\mathbf{x} \in \mathbb{R}^{3N}$ as the stacked vector of N vertices describing the quasistatic state of a fabric specimen, then $\mathbf{x}$ satisfies:

$$\begin{aligned}\mathbf{0} &= \mathbf{f}^{\text{stretch}}(\mathbf{x}) + \mathbf{f}^{\text{bend}}(\mathbf{x}) + \mathbf{f}^{\text{gravity}}(\mathbf{x}) \\ &= \mathbf{f}^{\text{stretch}}(\mathbf{x}) + \begin{bmatrix}\cdots & \mathbf{f}_m^{\text{bend}}(\mathbf{x}) & \cdots\end{bmatrix}\mathbf{k} + \rho\mathbf{f}_0^{\text{gravity}}(\mathbf{x}),\end{aligned} \tag{9}$$

Table 1. The performances of the GMM model and the VAE model. Here K is the number of Gaussian functions in the GMM model. The VAE-1 model uses four fully connected (FC) layers with 2,048, 1,024, 512 and 128 units, and a 64-dimensional latent space. The VAE-2 model uses four FC layers with 512, 256, 128 and 32 units, and a 32-dimensional latent space. The VAE-3 model uses three FC layers with 128, 32 and 16 units, and an 8-dimensional latent space. Finally, the VAE-4 model uses three FC layers with 32, 16 and 8 units, and a 4-dimensional latent space.

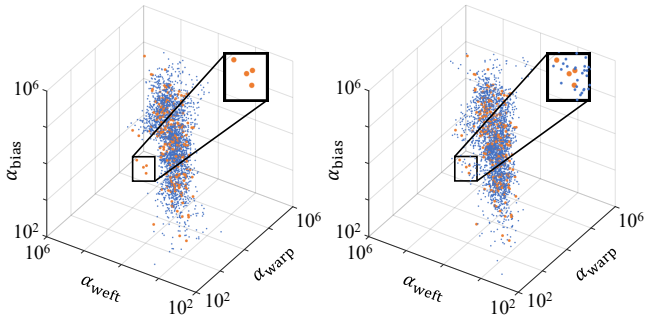
Model	Log likelihood (training)	Log likelihood (validation)	Wasserstein distance
GMM ($K = 3$)	-15.07	-14.86	98.9
GMM ($K = 4$)	-14.45	-13.88	84.6
GMM ($K = 5$)	-13.89	-13.61	72.6
GMM ($K = 6$)	-15.63	-15.71	72.5
GMM ($K = 7$)	-16.20	-15.79	97.6
VAE-1	-8.91	-8.68	14.3
VAE-2	-8.35	-7.92	13.9
VAE-3	**-6.91**	**-6.42**	**11.1**
VAE-4	-7.01	-6.59	12.4

in which $\mathbf{f}^{\text{stretch}}$, $\mathbf{f}^{\text{bend}}$ and $\mathbf{f}^{\text{gravity}}$ are the stretching, bending and gravitational force vectors, and $\mathbf{f}_m^{\text{bend}}(\mathbf{x})$ and $\mathbf{f}_0^{\text{gravity}}(\mathbf{x})$ are their linear components. Since we assume that the deformation is nearly isometric in cantilever and drape tests, $\mathbf{f}^{\text{stretch}}(\mathbf{x})$ can be scaled arbitrarily without affecting the drape, as long as it remains sufficiently large. As a result, $\mathbf{k}$ and ρ form a linear relationship, and $\mathbf{x}$ depends on $\mathbf{k}/\rho$, rather than the magnitude of $\mathbf{k}$. Intuitively, as the density ρ increases, we can increase bending parameters $\mathbf{k}$ by the same factor, to achieve the identical drape outcome as Fig. 14 shows.

Based on the aforementioned analysis, we assume that all of the fabrics share the same standard density $\bar{\rho} = 300\text{g}\cdot\text{m}^{-2}$ in our system. After we estimate the bending parameters of a specific fabric by our system, we can easily convert them into actual parameters by multiplying them with $\rho/\bar{\rho}$, in which the actual density ρ is measured by a scale.

5.1.3 The VAE subspace model. Once we collect the (normalized) parameter data set, our next job is to build a parameter subspace. Here we consider two popular choices: the Gaussian mixture model (GMM) [Duda and Hart 1973] and the variational autoencoder (VAE) [Kingma and Welling 2019] model. The idea behind the GMM model is to use a mixture of Gaussian functions to define probability densities in a subspace. To generate parameter vector samples under the GMM model, we just apply sampling to those mixed Gaussian functions. Differently, the VAE model takes a parameter vector as input and then tries to restore it in its output, through an encoder-decoder process. The structure of the decoder is typically the reverse of the structure of the encoder. Given the trained VAE model, we generate parameter vector samples by using the decoder to convert random latent space vectors to parameter sample outputs.

To evaluate these two models together with their parameter choices, we treat 80% of the real fabric data acquired in Subsection 5.1.1 as the training data set and the rest as the validation data set. We train the GMM model by the Expectation-Maximization



(a) The GMM model ($K = 5$) (b) The VAE-3 model

Fig. 15. A visual comparison of the parameter subspace models. Here each orange dot is the acquired parameter vector of a real-world fabric, and each blue dot is a newly generated parameter vector sample. The VAE-3 model covers the data in the dark box, but the GMM model fails to do so.

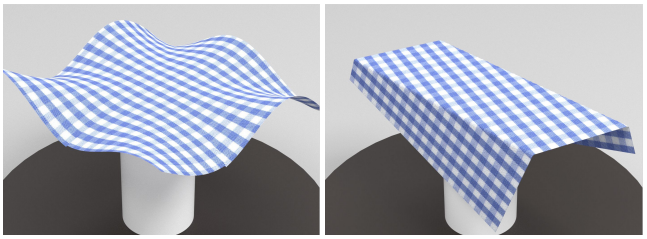
algorithm using 200 iterations; and we train the VAE model by an Adam optimizer, with the weight decay set to 10^{-5}, the learning rate set to 10^{-4}, and the batch size set to 16. Table 1 shows that the VAE-3 model with three fully connected layers and an 8-dimensional latent space outperforms other choices, in terms of both log-likelihood and Wasserstein distance metrics [Chizat et al. 2020]. Fig. 15 compares the difference between the GMM model ($K = 5$) and the VAE-3 model visually, using 6,000 newly generated parameter vector samples (in blue). Overall, the VAE-3 model can more correctly represent the underlying parameter subspace spanned by acquired bending parameters. Therefore, we choose it to be our subspace model.

There is one unique issue we must consider before we can use the learned subspace for sample generation: it may not be large enough to cover all of the real-world fabrics, due to insufficient training data and limited effectiveness of cantilever measurement. To address this issue, instead of sampling each latent space variable from a standard normal distribution $N(0, 1)$, we sample it from a Gaussian distribution $N(\mu, \sigma)$, where $\mu \in [-0.5, 0.5]$ and $\sigma \in [0.8, 1.2]$ are two uniformly distributed random variables. Doing this effectively enlarges the subspace covered by the generated vectors.

5.2 Data Synthesis

Given the bending parameter vectors randomly sampled from the learned subspace, we need to produce their corresponding multi-view depth images next. This can be divided into two steps: the generation of simulation data from the parameters, and the synthesis of multi-view depth images from simulation data. In both steps, we need data augmentation to address uncertainties existing in the real world.

5.2.1 Simulation data generation. To generate simulation data from parameter vector samples, we should make our simulation environment identical to the drape tester shown in Fig. 4. This includes defining the virtual specimen and the virtual platform to be the same sizes as the real ones, and specifying initial contact between the specimen and the platform with adhesive boundary conditions to account for static friction. But there exists an interesting question: *what is the initial shape of the specimen*? The initial shape plays an important role in determining the drape outcome, but human and environmental factors make it difficult to be accurately captured. From a mathematical perspective, the quasistatic simulation objective contains multiple local minima, and different initial shapes can lead to different minima solutions[2]. To address this problem, we enumerate all of the local minima by exploring different initial states for every parameter vector sample. In our system, we use three types of initial states.



(a) A type-I initial state (b) A type-II initial state

Fig. 16. Two types of initial states. For every parameter vector sample, we run our simulator multiple times with different initial states, so as to address the local minima issue in our simulation data set.

- The first type of an initial state is made by adding random sine waves to a flat surface, as shown in Fig. 16a.
- The second type of an initial state is made by purposefully folding the specimen along a randomly selected direction, as Fig. 16b shows.
- Finally, we randomly choose a simulated drape mesh of another parameter vector sample in the existing data set as an initial state of the current sample.

For every parameter vector sample, we create eight initial states at random. We then add a small drift to each initial state, to account for imperfect alignment between the specimen center and the cylinder center in the real world. Each initial state is represented by a 100×100 cloth mesh, with its triangulation slightly perturbed to make the data set more robust against mesh topological changes [Yang et al. 2017]. Given the mesh of each initial state, we run our simulator until it reaches quasistatic equilibrium, typically under twenty seconds. Once we obtain the eight drape mesh outcomes corresponding to the eight initial states, we add them together with their common parameter vector into the simulation data set.

5.2.2 Depth image synthesis. Next we need to convert simulation data into multi-view depth images. When using the drape tester in Subsection 3.1, we face three main sources of uncertainties in captured depth images: the unknown orientation of the fabric specimen, the unknown exact camera configuration due to calibration errors, and depth sensor noise and errors. To counteract such uncertainties, we use stratified sampling to obtain 12 random orientations for every simulated drape model. We then use these 12 orientations, together with random perturbations to camera position/orientation/field of view, to synthesize 12 sets of multi-view depth images at the resolution of 240×180. Each set of multi-view

[2]The local minima issue is not the only cause of multiple drape outcomes. Another cause is hysteresis, mainly due to internal friction.

(a) The front side (b) The left side (c) The back side (d) The right side

Fig. 17. Multi-view depth images. Each depth image encodes the shape of the same drape in a different view. We treat these depth images in multiple views together as the feature vector of one data point.

Table 2. The performances of different backbone networks. This table shows that ResNet-18 is an optimal choice in terms of both the training time and the RMSE errors.

Name	Training time (hrs)	Number of variables	RMSE (training)	RMSE (validation)
VGG16	100.8	105M	0.153	**0.122**
AlexNet	**12.4**	46M	0.252	0.224
ResNet-18	12.7	13M	**0.147**	0.127
EfficientNet B0	27.0	4M	0.187	0.142

depth images contains four views as Fig. 17 shows. Finally, we apply Kinect-related noise [Barron and Malik 2013; Bohg et al. 2014] and affine transformation perturbations to synthetic depth images, so they can be more similar to real depth images with noise and errors. We treat each set and the corresponding parameter vector as one data point in the synthetic data set.

In total, we generate 6,000 parameter vector samples and simulate 48,000 drape models using the methodology in Subsection 5.2.1. The whole simulation process takes approximately 12 hours on a computer cluster with 22 CPUs. We then convert the simulated drapes into the synthetic depth image data set, which contains $48,000 \times 12 \times 4 = 2.3$M depth images and takes approximately two hours to synthesize. The total storage cost is 12GB.

6 NEURAL NETWORK DESIGN

As shown in Fig. 3, our neural network takes multi-view depth images as input and infers bending stiffness parameters as output. While the concept of this network is simple, there are many design choices involved in it. In this section, we will focus our research on the evaluation of these choices. Our evaluation metric, also acting as the loss function for training, is the RMSE error between the normalized ground truth $\{\mathbf{g}_i\}$ and the predictions $\{\mathbf{p}_i\}$:

$$\mathcal{L} = \left(\frac{\sum_{i=1}^{I} \|\mathbf{g}_i - \mathbf{p}_i\|^2}{I} \right)^{1/2}, \tag{10}$$

in which I is the batch size. By default, we use 80% of the synthetic data for training and keep the rest for validation.

6.1 The Choice of the Backbone Network

The first and foremost choice is the backbone of our neural network. Table 2 compares the performances of multiple backbone choices, including VGG16 [Rasheed et al. 2020; Simonyan and Zisserman 2015], AlexNet [Krizhevsky et al. 2012], ResNet-18 [He et al. 2016] and EfficientNet B0 [Tan and Le 2019]. When training each network, we make a good faith effort to tune its hyper-parameters for optimal validation loss. According to Table 2, ResNet-18 is the best choice with the shortest training time, so we choose to use it in the rest of this paper.

The training of our ResNet-18 backbone uses an Adam optimizer, with the weight decay set to 10^{-4}, the learning rate set to 10^{-4}, the batch size set to 128 and the learning rate decay set to 0.995. The training process runs on a workstation with 96 Intel CPU cores and four NVIDIA GeForce RTX 3090 GPUs.

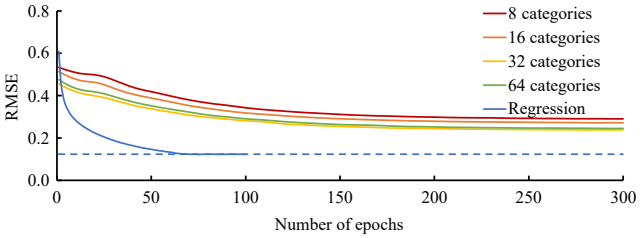


Fig. 18. The learning curves of classification and regression neural networks. When we define parameter estimation as a classification problem, we need a large number of categories to reduce the discretization error. But doing so requires a large amount of training data that we cannot afford. Overall, it is more effective to use a regression neural network as in our system.

6.2 The Choice of the Output

We can use either a regression neural network to output bending stiffness parameters directly, or a classification neural network to classify fabrics into predefined bending stiffness groups. To know which one is better, we build both neural networks. Specifically, we follow [Yang et al. 2017] to build a classification network, by choosing one fabric as the basis and discretizing each continuous parameter space into 8, 16, 32 or 64 categories. In our experiment, the basis is the polar fleece fabric with moderate and nearly isotropic bending stiffness.

Fig. 18 compares the validation RMSE errors of both classification and regression networks, and it indicates that a regression network is a better choice for our system. The main reason is because a classification neural network needs a sufficiently large number of categories to reduce the discretization error. But as the number of categories grows beyond 32, the RMSE error actually increases due to insufficient data for training a fine-grained classification network. We note that the computational cost of training a classification network is also typically higher.

6.3 The Number of Views

Our next question is: how many depth image views we should incorporate into the feature vector input? Ideally, the number of views should be large enough to cover all of the drape shape details. On the other hand, too many views would increase the difficulty of training a network and eventually undermine the estimation quality. According to our experiment in Fig. 19, the optimal choice is to use four depth image views with an even separation of $\pi/2$.

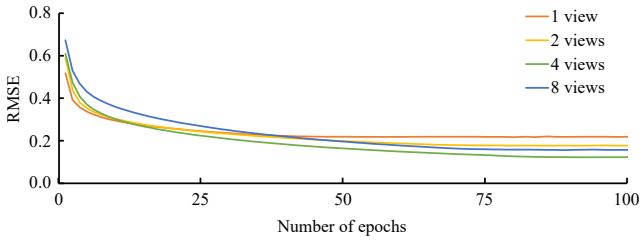


Fig. 19. The learning curves when using different numbers of depth image views in the feature vector input. They show that the validation RMSE error drops as the number of views increases from one to four. But as the number of views increases from four to eight, the error starts to grow, due to increased difficulty of training a network.

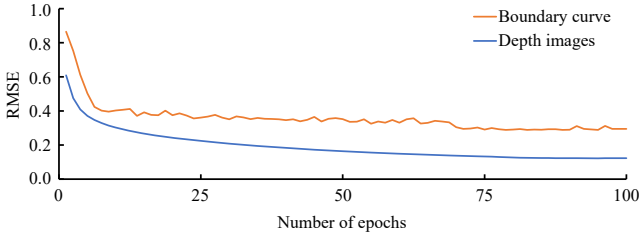


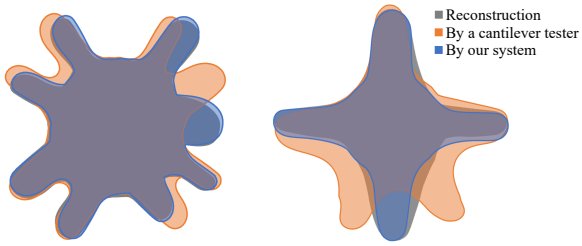
Fig. 20. The learning curves when using different feature vectors. By taking multi-view depth images as input, our neural network more accurately estimates bending parameters according to the validation RMSE error.

6.4 The Choice of the Feature Vector

Finally, we evaluate two options for the feature vector input: the boundary curve of the drape as in [Ju and Choi 2020] and the multi-view depth images of the drape as proposed in this paper. Compared with the boundary curve, the depth images are not only more accessible, but also more capable of expressing shape details, especially the interior not covered by the boundary curve. In this experiment, we train a network that takes the boundary curve as input, using the same loss function and network parameters given in [Ju and Choi 2020]. Fig. 20 shows the RMSE error of using the depth image input is lower than that of using the boundary curve input.

7 RESULTS AND DISCUSSIONS

(Please refer to the supplemental video and documents for detailed examples.) In this section, we would like to comprehensively study the performance of our learning-based parameter estimation system, using 32 woven and knitted fabrics newly selected from a fabric inventory. These fabrics are commonly used for making functional and fashionable garments in the real world, and they are not the same ones used for constructing the parameter subspace in Subsection 5.1.1. Their compositions include natural and synthetic materials, and they exhibit a wide range of physical properties, from light (with $\rho = 55\text{g}\cdot\text{m}^{-2}$) to heavy (with $\rho = 480\text{g}\cdot\text{m}^{-2}$), from soft (with $\alpha = 548\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$) to stiff (with $\alpha = 137,063\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$), from isotropic to anisotropic, and from nearly linear to highly nonlinear. According to the estimations, fabric #25 demonstrates highly anisotropic bending stiffness, with $\alpha^{\text{warp}} = 98,849\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$, $\alpha^{\text{bias}} = 23,468\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$ and $\alpha^{\text{weft}} = 21,192\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$. Meanwhile, fabric #21 demonstrates the highest nonlinearity, with $\alpha^{\text{warp}} = 136,122\text{g}\cdot\text{mm}^2\cdot\text{s}^{-2}$ and $\beta^{\text{warp}} = -4,933,507\text{g}\cdot\text{mm}^3\cdot\text{s}^{-2}$. As mentioned before, we choose the nonlinear dihedral angle model by default. Please check the supplemental catalog for more fabric property details.



(a) Fabric #9 (b) Fabric #27

Fig. 21. The top-view silhouettes of the simulated drapes. Compared with the simulated drapes (in orange) using a cantilever tester, the simulated drapes (in blue) using our system are more similar to the reconstructed shapes (in gray).

Next we will evaluate the performance of our system from four perspectives: the cost, the simulation fidelity, the reliability, and finally the applicability to different use cases.

7.1 Cost Analysis

After we train the neural network, there are two costs associated with the use of our system: the time cost and the financial cost.

According to our experiment, it takes under three minutes for an inexperienced user to complete the whole parameter estimation process of one fabric using our system. In these three minutes, the user needs approximately two minutes to prepare the square fabric specimen, and another minute to capture depth images and make bending parameter inferences. We note that without a special tool, a square specimen is much easier to cut than a circular one used in the original Cusick drape method.

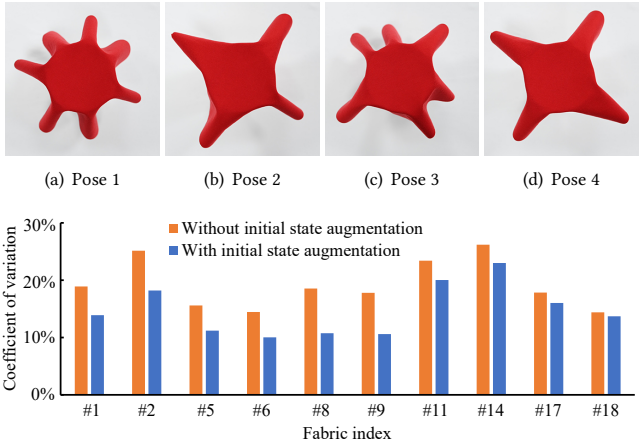
Our drape tester is inexpensive and easy to build. Besides the Azure Kinect depth camera, the rest of the device costs under 30 US dollars and it takes little time to assemble. In comparison, the parts used by our cantilever tester are customized and the cantilever tester (without the DSLR camera) costs at least 500 US dollars.

7.2 Simulation Fidelity Analysis

In Section 6, we use the synthetic data set to validate the effectiveness of the trained network in finding suitable bending parameters. In this subsection, we would like to further evaluate the effectiveness of our network in handling real fabrics. Since their ground truth parameters are unknown, we cannot calculate the RMSE error directly. Instead we test if the estimated parameters can help the simulator reproduce the captured drape shape closely. To do so, we use a 3D surface scanner to reconstruct the drape of a real fabric specimen as an initial state, run the simulator until the residual force becomes sufficiently small, and then check how far the quasistatic equilibrium state departs from the initial state. We note that this

Table 3. Quantitative evaluation on the fidelity of our simulator when it uses the bending parameters estimated in different ways. Here the two metrics are: the mean vertex displacement (in mm) between the captured shape and the simulated shape, and the mean of the relative mean curvature difference (in parentheses). The table shows that the parameters estimated by our system with the nonlinear dihedral angle model are generally the optimal ones.

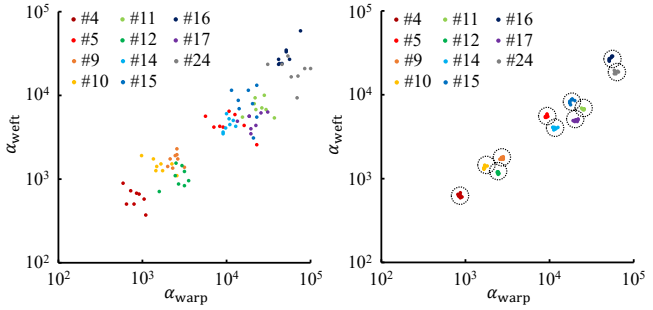
Fabric index	Cantilever	Classification	GMM	VAE		
	nonlinear dihedral	[Yang et al. 2017]	nonlinear dihedral	linear quadratic	linear dihedral	nonlinear dihedral (ours)
#1	1.3 (8.0%)	1.8 (12.7%)	**1.2** (8.0%)	1.3 (7.0%)	1.6 (7.6%)	1.3 (**5.6%**)
#5	1.9 (14.6%)	4.6 (8.8%)	1.9 (6.6%)	1.7 (5.8%)	1.4 (5.2%)	**0.9** (**3.3%**)
#7	2.5 (14.5%)	2.7 (18.0%)	2.7 (13.7%)	1.9 (7.5%)	1.7 (12.0%)	**1.7** (**4.4%**)
#9	1.4 (4.9%)	3.5 (8.9%)	1.1 (1.5%)	1.1 (**1.1%**)	**0.8** (1.3%)	1.0 (1.2%)
#11	1.3 (11.0%)	4.6 (18.6%)	1.2 (14.3%)	1.6 (6.6%)	1.1 (**6.5%**)	**1.0** (7.4%)
#12	2.6 (6.6%)	4.3 (6.6%)	2.3 (7.5%)	1.7 (6.8%)	1.1 (6.1%)	**0.7** (**5.9%**)
#24	1.4 (6.9%)	2.9 (14.2%)	0.9 (5.5%)	0.9 (5.8%)	1.1 (6.6%)	**0.7** (**3.5%**)
#27	2.8 (5.7%)	2.9 (10.6%)	1.9 (4.7%)	1.9 (3.1%)	2.0 (**2.6%**)	**1.7** (2.7%)
#28	3.2 (6.8%)	3.3 (8.9%)	2.5 (6.9%)	2.7 (3.9%)	**2.2** (3.5%)	2.6 (**3.3%**)
#29	2.8 (6.4%)	4.7 (7.2%)	2.3 (4.1%)	2.1 (3.3%)	2.5 (3.4%)	**2.0** (**2.8%**)



(a) Pose 1 (b) Pose 2 (c) Pose 3 (d) Pose 4

(e) The coefficients of variation

Fig. 22. The reliability of our system with respect to drape shape variation. In the real world, we can pose the same fabric specimen into different drapes as shown from (a) to (d). In spite of that, our network makes reliable estimations with low coefficients of variation, thanks to the data augmentation approach by using multiple initial states.



(a) Without data augmentation (b) With data augmentation

Fig. 23. The reliability of our system with respect to initial orientation. Thanks to data augmentation, our system reliably estimates bending parameters without requiring the specimen to be placed in any specific orientation.

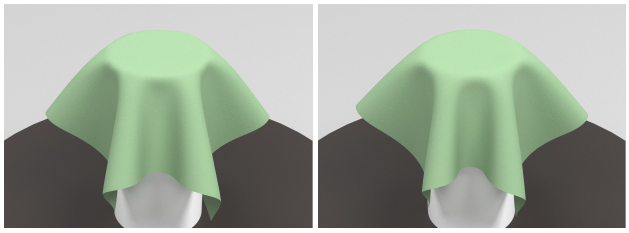
practice also circumvents the local minima issue associated with quasistatic simulation in Subsection 5.2.1.

Table 3 summarizes the difference between the reconstructed initial state and the simulated quasistatic equilibrium state by two metrics: the mean vertex displacement (in mm) and the mean of the relative mean curvature difference. It evaluates the bending parameters estimated by various approaches, including a cantilever tester (as described in Subsection 4.3), the video-based classification method implemented by [Yang et al. 2017], our system using the GMM subspace and our system using the VAE subspace with multiple anisotropic bending models. Regarding the simulation fidelity, this experiment shows that our system outperforms a cantilever tester and the video-based method in general. It also shows that the use of a nonlinear dihedral angle model allows our simulator to better reproduce the captured drape shape, as expected. Fig. 21 illustrates the difference between the reconstructed state and the simulation state for two fabrics in a top view.

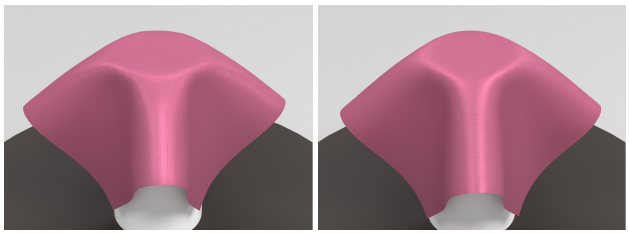
7.3 Sensitivity Analysis

In this subsection, we test if our system can make reliable parameter estimation of the same real-world fabric, when the vector input varies for various reasons.

7.3.1 Sensitivity to drape shape variation. In the real world, we can manually pose the same fabric specimen into different drape shapes as shown in Fig. 22a to 22d, due to the existence of local minima and hysteresis effects. Fig. 22e shows that while the drape shape input varies, our system is able to make reliable parameter estimation with a low coefficient of variation. This reliability depends on the use of multiple initial states during our data augmentation process, as discussed in Subsection 5.2. Without it, the coefficient of variation becomes approximately five percent greater. Having said that, the variation of our estimation is still not negligible, especially for Fabric #14. We believe this is because our approach addresses only the local minima issue, but not hysteresis.



(a) Our simulator (quadratic) (b) ARCSim (quadratic)



(c) Our simulator (nonlinear dihedral) (d) Argus (nonlinear dihedral)

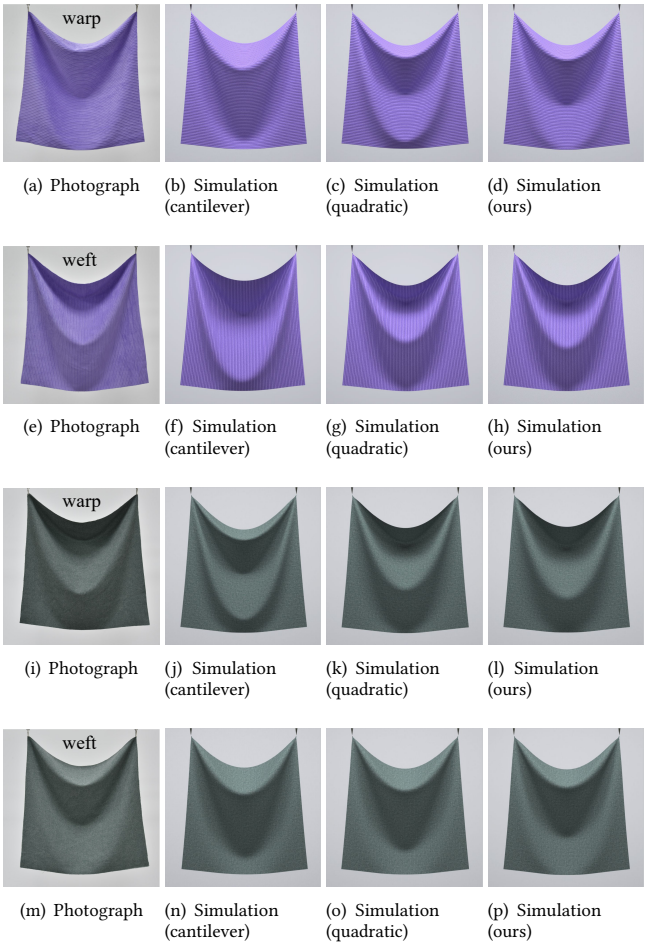
Fig. 24. The drapes simulated by our simulator, ARCSim [Narain et al. 2012] and Argus [Li et al. 2018]. This example shows that different simulators can produce almost identical drape outcomes, as long as the underlying bending models and their parameters are the same.

7.3.2 Sensitivity to fabric orientation. Unlike [Ju and Choi 2020], our system does not require a fabric specimen to be placed on the drape tester in any specific orientation. This is because we augment our training data by synthetic depth images in random initial orientations (as discussed in Subsection 5.2.2) and our network should be able to make reliable inference with no restriction on initial orientation. To verify this strength, we use our network to estimate bending parameters of the same fabric specimen multiple times, but in different initial orientations. Fig. 23b shows that our network is indeed reliable regardless of initial orientation. Without such data augmentation, the estimation becomes unreliable as shown in Fig. 23a.

7.4 Applicability Analysis

While we have demonstrated that the estimated parameters effectively help our own simulator achieve high fidelity in drape tests as Subsection 7.2 shows, we want to know if other simulators can achieve such fidelity in other environments as well.

7.4.1 Applicability to other simulators. First we would like to evaluate the applicability of our system to other cloth simulators, such as ARCSim [Narain et al. 2012] and Argus [Li et al. 2018]. To make this evaluation meaningful, we integrate the quadratic bending model into ARCSim and the nonlinear dihedral angle model into Argus. Fig. 24 shows that the drapes simulated by ARCSim and Argus are indeed nearly identical to ours (with the maximal vertex displacement under 0.1mm) when the bending models and their parameters are the same. From this experiment, we can draw two conclusions: the estimated parameters also improve the fidelity of other simulators; and our system does not rely on our own simulator for data generation and it can adopt other simulators as well.



(a) Photograph (b) Simulation (cantilever) (c) Simulation (quadratic) (d) Simulation (ours)

(e) Photograph (f) Simulation (cantilever) (g) Simulation (quadratic) (h) Simulation (ours)

(i) Photograph (j) Simulation (cantilever) (k) Simulation (quadratic) (l) Simulation (ours)

(m) Photograph (n) Simulation (cantilever) (o) Simulation (quadratic) (p) Simulation (ours)

Fig. 25. Hanging test examples. We drape a large 600mm×600mm fabric specimen by lifting up its corners, and we compare the drapes in photograph and simulation. Being aware of the multiple shape issue, we iron each fabric, hang them several times and choose the most relaxed one as the ground truth. The figure shows that the parameters estimated by our system are applicable to hanging cases as well. In this figure, we evaluate fabric #18 and fabric #19.

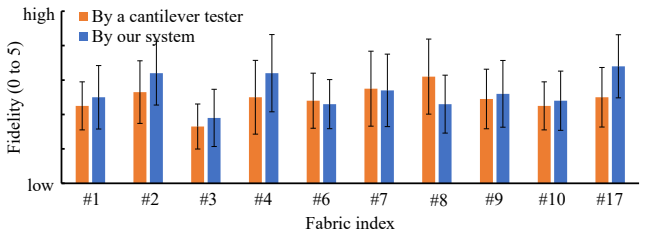
7.4.2 Applicability to the hanging test. Similar to [Wang et al. 2011], we use the hanging test to evaluate the usability of our parameters in a different draping environment. In this test, we lift a 600mm×600mm fabric specimen by its two corners, with a separation distance of 500mm as Fig. 25 shows. Meanwhile, we simulate a cloth mesh with 200×200 vertices, using the nonlinear dihedral bending parameters estimated by a cantilever tester and by our system. Similar to the Cusick drape test, the hanging drape test also suffers from the multiple shape issue. To eliminate its influence on evaluation, we manually pose each fabric drape in a sufficiently unique and relaxed way with minimal folds. Fig. 25 shows the simulation using our parameters is more similar to the photograph.

7.4.3 Applicability to the skirt test. Next we want to know how the estimated bending parameters affect wrinkles on simulated

(a) Simulation (cantilever) (b) Photograph (c) Simulation (ours)

(d) Simulation (cantilever) (e) Photograph (f) Simulation (ours)

(g) Simulation (cantilever) (h) Photograph (i) Simulation (ours)



(j) Means and standard deviations of user assessments

Fig. 26. Skirt test examples. In the skirt test, we invite potential users to assess our simulation results together with the photographs of real skirts made by ten different fabrics, such as fabric #2, #4 and #17 in (b), (e) and (h). In general, the users concur that the effectiveness of our system outperforms that of a cantilever tester as shown in (j).

garments, rather than simulated drape specimens. To answer this question, we design a skirt test shown in Fig. 26. Our skirt model contains 40K vertices and 80K triangles, and its size parameters are: 753mm waist, 2,629mm bottom and 310mm height. We craft ten such skirts out of different fabrics and place them on a female mannequin with a 731mm waist and a 899mm hip. We then run our simulator to simulate these skirts, with the bending parameters estimated by a cantilever tester and by our system respectively. With IRB approval, we invite 40 future users of our system, including fashion designers, pattern makers and technical artists, to assess the fidelity of rendered skirt images in random order. We inform

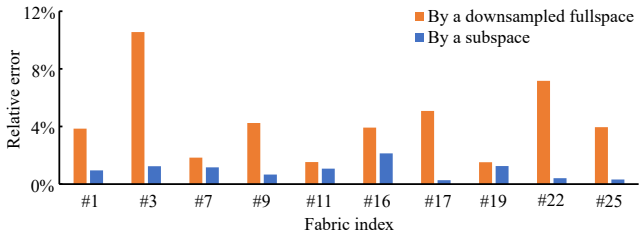


Fig. 27. The prediction differences between trained networks. We train a network using synthetic data from a full-space sample set and treat its predictions as ground truths. We then train two other networks, using a subspace sample set and a downsampled set, respectively. This figure shows that the use of the parameter subspace effectively reduces the computational time while preserves the prediction accuracy.

the users that the goal is to evaluate if simulated bending effects are similar to those in photographs, rather than to make a direct comparison between wrinkle details. This user study indicates that the difference between the use of a cantilever tester and the use of our system is not so significant in the skirt test, and the users think all of the results have space for improvement. But in general, the users agree that our system is slightly more effective as illustrated in Fig. 26j.

7.5 The Necessity of A Parameter Subspace

The parameter subspace in Subsection 5.1 plays an important role in the affordability of our synthetic data acquisition process. In the following experiment, we justify its necessity, by showing its impact on the prediction accuracy of our network. To do so, we directly sample every linear bending parameter 28 times and collect $28^3 = 22$K parameter vector samples in the full three-dimensional space. Since such a sample set is already large, we do not consider nonlinearity or data augmentation. We generate synthetic data using this sample set, train our network and use the network to make predictions as ground truths. We then collect another set of 3,650 parameters but in the subspace this time, and train another network without nonlinearity or data augmentation again. Fig. 27 shows that the prediction difference between these two networks is small, while the computational time after using the subspace is only one sixth. Fig. 27 also shows that simply downsampling the full-space sample set to a subset with 3,650 samples would significantly downgrade the prediction accuracy.

7.6 A Long Dress Example

Finally, we assess the use of our estimated bending parameters in dynamic simulation of a complex garment with wrinkle details, as Fig. 1 shows. In this experiment, we simulate the long dress model multiple times, with the parameters estimated from fabric #1, #9, #10, #17, #20 and #24 respectively. The mesh of the dress model contains 116K vertices and 226K triangles, and our simulator can simulate one frame approximately every 0.08 seconds at the time step $\Delta t = 0.01$s. The whole animation sequence contains 1,000 time steps. Thanks to our bending stiffness parameters, the simulator can animate the model with a variety of realistic wrinkle details. We note that body movement can occasionally stretch the dress. To

simulate stretching behaviors of a fabric realistically as well, we measure its planar stiffness parameters by a tensile tester, which is beyond the scope of this research.

7.7 Limitations

The greatest limitations of our learning-based parameter estimation system are from the inaccuracy of hardware and the limited capability of cloth simulation. Commodity depth cameras suffer from many noise issues and the design of our system must find a balance between affordability and accuracy. In fact, we have experimented with nearly ten depth cameras and decided to use Azure Kinect in the end. Meanwhile, there are many accuracy and expressiveness issues involved in physics-based cloth simulation and we do not expect to eliminate them soon. The problem is that if the capability of a simulator is limited, then bending behaviors of certain fabrics cannot be correctly simulated, no matter how we tune the parameters. In particular, our simulator and our system fail to consider hysteresis, which is now a main source of errors in simulation, estimation and evaluation. Our current system assumes that the same simulator is used for both data generation and animation production. This practice allows the system to take the inaccuracy of our simulator into consideration, but it also means the estimated parameters are less suitable for other simulators, unless they use similar bending models as Subsection 7.4.1 shows. Finally, the system does not consider complex bending behaviors, such as those caused by double-sided fabrics. The curly boundary issue also exists in the drape test, but it is not so problematic as in the cantilever test.

8 CONCLUSIONS AND FUTURE WORK

Based on the Cusick drape test method, we justify the effective use of deep neural networks in the development of a novel bending stiffness parameter estimation system. Our experiment indicates that the resulting drape test outperforms cantilever tests in terms of reliability, effectiveness and usability. The experiment also reveals that the bending behaviors of real-world fabrics are complex and the estimation of bending stiffness needs further investigation.

As discussed in Subsection 7.7, we plan to address the effectiveness of our system from both hardware and software perspectives. We are committed to improving our hardware designs. We will also study the importance of hysteresis effects (largely due to internal friction) and how to address them in both estimation and simulation. In the long term, we plan to develop a unified test method for estimating planar and bending stiffness at the same time, using the same fabric specimen. This will not only reduce the test time, but also allow us to explore complex fabric behaviors not considered by the current system, such as bending under large stretching.

ACKNOWLEDGMENTS

We wish to thank the digital content team at Style3D for helping with data collection and animation production. We also would like to thank NVIDIA for hardware support. This work is partially funded by NSFC 61732016.

REFERENCES

ASTM. 2016. ASTM D4032: Standard Test Method for Stiffness of Fabric by the Circular Bend Procedure. (Dec. 2016).

ASTM. 2018. ASTM D1388: Standard Test Method for Stiffness of Fabrics. (July 2018).

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of SIGGRAPH 98 (Computer Graphics Proceedings, Annual Conference Series)*, Eugene Fiume (Ed.). ACM, 43–54.

Jonathan T. Barron and Jitendra Malik. 2013. Intrinsic Scene Properties from a Single RGB-D Image. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 17–24.

Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Proceedings of SGP* (Cagliari, Sardinia, Italy). 227–230.

Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. 2003. Estimating Cloth Simulation Parameters from Video. In *Proceedings of SCA* (San Diego, California). 37–51.

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. 2009. Capture and Modeling of Non-linear Heterogeneous Soft Tissue. *ACM Trans. Graph. (SIGGRAPH)* 28, 3, Article 89 (July 2009), 9 pages.

Jeannette Bohg, Javier Romero, Alexander Herzog, and Stefan Schaal. 2014. Robot Arm Pose Estimation through Pixel-Wise Part Classification. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 3143–3150.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph. (SIGGRAPH)* 33, 4, Article 154 (July 2014), 11 pages.

Katherine L. Bouman, Bei Xiao, Peter Battaglia, and William T. Freeman. 2013. Estimating the Material Properties of Fabric from Video. In *2013 IEEE International Conference on Computer Vision*. 1984–1991.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (July 2002), 594–603.

Robert Bridson, Sebastian Marino, and Ronald Fedkiw. 2003. Simulation of Clothing with Folds and Wrinkles. In *Proceedings of SCA* (San Diego, California). 28–36.

Enric Carrera-Gallissa, Xavier Capdevila, and Josep Valldeperas. 2017. Evaluating Drape Shape in Woven Fabrics. *The Journal of the Textile Institute* 108, 3 (2017), 325–336.

Satyasaran Changdar, Bivas Bhaumik, and Soumen De. 2021. Physics-Based Smart Model for Prediction of Viscosity of Nanofluids Containing Nanoparticles Using Deep Learning. *Journal of Computational Design and Engineering* 8, 2 (2021), 600–614.

Zhili Chen, Renguo Feng, and Huamin Wang. 2013. Modeling Friction and Air Effects between Cloth and Deformable Bodies. *ACM Trans. Graph. (SIGGRAPH)* 32, 4 (July 2013), 8 pages.

Lénaïc Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. 2020. Faster Wasserstein Distance Estimation with the Sinkhorn Divergence. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada). Article 190, 13 pages.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (July 2002), 604–611.

CLO. 2022. Fabric Kit Manual. https://support.clo3d.com/hc/en-us/articles/360041074334-Fabric-Kit-Manual.

David Clyde, Joseph Teran, and Rasmus Tamstorf. 2017. Modeling and Data-Driven Parameter Estimation for Woven Fabrics. In *Proceedings of SCA* (Los Angeles, California). Article 17, 11 pages.

Gordon E. Cusick. 1965. The Dependence of Fabric Drape on Bending and Shear Stiffness. *Journal of the Textile Institute* 56, 11 (1965), 596–606.

Abe Davis, Katherine L. Bouman, Justin G. Chen, Michael Rubinstein, Oral Büyüköztürk, Frédo Durand, and William T. Freeman. 2017. Visual Vibrometry: Estimating Material Properties from Small Motions in Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2017), 732–745.

Richard O. Duda and Peter E. Hart. 1973. *Pattern Classification and Scene Analysis*. A Wiley-Interscience publication.

Elliot English and Robert Bridson. 2008. Animating Developable Surfaces Using Nonconforming Elements. *ACM Trans. Graph. (SIGGRAPH)* 27, 3 (Aug. 2008), 1–5.

Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-Seidel Method for Stable Soft Body Dynamics. *ACM Trans. Graph. (SIGGRAPH Asia)* 35, 6, Article 214 (Nov. 2016), 9 pages.

Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. 2019. Deep Inverse Rendering for High-Resolution SVBRDF Estimation from an Arbitrary Number of Images. *ACM Trans. Graph. (SIGGRAPH)* 38, 4, Article 134 (July 2019), 15 pages.

Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of SCA* (San Diego, California). 62–67.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

ISO. 2008. ISO 9073-9: Textiles – Test methods for nonwovens – Part 9: Determination of drapability including drape coefficient. (March 2008).

Eunjung Ju and Myung Geol Choi. 2020. Estimating Cloth Simulation Parameters from a Static Drape Using Neural Networks. *IEEE Access* 8 (2020), 195113–195121.

Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2008. Simulating Knitted Cloth at the Yarn Level. *ACM Trans. Graph. (SIGGRAPH)* 27, 3, Article 65 (Aug. 2008), 9 pages.

Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2010. Efficient Yarn-Based Cloth with Adaptive Contact Linearization. *ACM Trans. Graph. (SIGGRAPH)* 29, 4, Article 105 (July 2010), 10 pages.

Kaizhang Kang, Minyi Gu, Cihui Xie, Xuanda Yang, Hongzhi Wu, and Kun Zhou. 2021. Neural Reflectance Capture in the View-Illumination Domain. *IEEE Transactions on Visualization and Computer Graphics* 1 (2021), 1–1.

Kato Tech. 2022. KES-FB2-AUTO-A Pure Bending Tester. https://english.keskato.co.jp.

Diederik P. Kingma and Max Welling. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends in Machine Learning* 12, 4 (2019), 307–392.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada). 1097–1105.

Shoji Kunitomo, Shinsuke Nakamura, and Shigeo Morishima. 2010. Optimization of Cloth Simulation Parameters by Considering Static and Dynamic Features. In *ACM SIGGRAPH 2010 Posters*. Article 15, 1 pages.

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E. Brown, and Laurence Boissieux. 2018. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH)* 37, 4, Article 52 (July 2018), 15 pages.

Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-Spring Systems. *ACM Trans. Graph. (SIGGRAPH Asia)* 32, 6, Article 214 (Nov. 2013), 7 pages.

Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse Elastic Shell Design with Contact and Friction. *ACM Trans. Graph. (SIGGRAPH Asia)* 37, 6, Article 201 (Dec. 2018), 16 pages.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. 2003. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*. Springer, 35–57.

Microsoft. 2022. Azure Kinect DK. https://azure.microsoft.com/en-us/services/kinect-dk.

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Comput. Graph. Forum (Eurographics)* 31, 2 (May 2012), 519–528.

Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C. Schvartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A. Otaduy. 2013. Modeling and Estimation of Internal Friction in Cloth. *ACM Trans. Graph. (SIGGRAPH Asia)* 32, 6, Article 212 (Nov. 2013), 10 pages.

Vishnu Mohan M S and Vivek Menon. 2021. Measuring Viscosity of Fluids: A Deep Learning Approach Using a CNN-RNN Architecture. In *The First International Conference on AI-ML-Systems (AIMLSystems 2021)*. Article 12, 5 pages.

Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. 2005. Meshless Deformations Based on Shape Matching. *ACM Trans. Graph. (SIGGRAPH)* 24, 3 (July 2005), 471–478.

Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH Asia)* 31, 6, Article 152 (Nov. 2012), 10 pages.

Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. 2001. Scanning Physical Interaction Behavior of 3D Objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH 2001)*. 87–96.

Dinesh K. Pai, Austin Rothwell, Pearson Wyder-Hodge, Alistair Wick, Ye Fan, Egor Larionov, Darcy Harrison, Debanga Raj Neog, and Cole Shing. 2018. The Human Touch: Measuring Contact with Real Human Soft Tissues. *ACM Trans. Graph. (SIGGRAPH)* 37, 4, Article 58 (July 2018), 12 pages.

Abdullah-Haroon Rasheed, Victor Romero, Florence Bertails-Descoubes, Stefanie Wuhrer, Jean-Sébastien Franco, and Arnaud Lazarus. 2020. Learning to Measure the Static Friction Coefficient in Cloth Contact. In *IEEE Conference on Computer Vision and Pattern Recognition* (Seattle, United States). 9909–9918.

Abdullah Haroon Rasheed, Victor Romero, Florence Bertails-Descoubes, Stefanie Wuhrer, Jean-Sebastien Franco, and Arnaud Lazarus. 2021. A Visual Approach to Measure Cloth-Body and Cloth-Cloth Friction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1.

Victor Romero, Mickaël Ly, Abdullah Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. 2021. Physical Validation of Simulators in Computer Graphics: A New Framework Dedicated to Slender Elastic Structures and Frictional Contact. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 66 (July 2021), 19 pages.

Ribeiro Rui, André Pilastri, Carla Moura, Filipe Rodrigues, Rita Rocha, Morgado José, and Cortez Paulo. 2020. Predicting Physical Properties of Woven Fabrics via Automated Machine Learning and Textile Design and Finishing Features. In *Artificial Intelligence Applications and Innovations*, Vol. 584. 244–255.

Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. 2009. Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction. *IEEE Transactions on Visualization and Computer Graphics* 15, 2 (March 2009), 339–350.

Shirin Shahriari, Gholamreza Pazuki, and Bandar Duraya Al-Anazi. 2010. Neutral Network Estimates Poisson's Ratio, Young's Modulus. *Oil and Gas Journal* 108, 19 (2010), 47–50.

Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.). 14 pages.

Taber Industries. 2022. Cantilever Tester - Model 145. http://www.ordertaber.com.

Rasmus Tamstorf and Eitan Grinspun. 2013. Discrete Bending Forces and Their Jacobians. *Graphical Models* 75, 6 (2013), 362–370.

Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML) 2019*, Vol. 97. PMLR, 6105–6114.

Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH Asia)* 37, 6 (Dec. 2018), 10 pages.

Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. 2009. A Simple Approach to Nonlinear Tensile Stiffness for Accurate Cloth Simulation. *ACM Trans. Graph.* 28, 4, Article 105 (Sept. 2009), 16 pages.

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation Capture and Modeling of Soft Objects. *ACM Trans. Graph. (SIGGRAPH)* 34, 4, Article 94 (July 2015), 12 pages.

Huamin Wang. 2015. A Chebyshev Semi-Iterative Approach for Accelerating Projective and Position-Based Dynamics. *ACM Trans. Graph. (SIGGRAPH Asia)* 34, 6, Article 246 (Oct. 2015), 9 pages.

Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. 2011. Data-Driven Elastic Models for Cloth: Modeling and Measurement. *ACM Trans. Graph. (SIGGRAPH)* 30, 4, Article 71 (July 2011), 9 pages.

Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Trans. Graph. (SIGGRAPH Asia)* 35, 6, Article 212 (Nov. 2016), 10 pages.

Longhua Wu, Botao Wu, Yin Yang, and Huamin Wang. 2020. A Safe and Fast Repulsion Method for GPU-Based Cloth Self Collisions. *ACM Trans. Graph.* 40, 1, Article 5 (Dec. 2020), 18 pages.

Shan Yang, Junbang Liang, and Ming C. Lin. 2017. Learning-Based Cloth Material Recovery from Video. In *IEEE International Conference on Computer Vision*. 4393–4403.