CHAPTER 6. CASUAL CAPTURE OF FABRIC MECHANICS

Figure 6.1: From two depth images of a fabric sample casually placed in two specific configurations (input), we infer its corresponding set of mechanical parameters. These can be used in a simulator, to visualize the drape of any garment (output). Our method introduces a perceptual drape similarity metric, which enables sorting materials based on their drape.

In this chapter, we propose a method to estimate the mechanical parameters of fabrics using a casual capture setup with a depth camera. Our approach enables to create mechanically-correct digital representations of real-world textile materials, which is a fundamental step for many interactive design and engineering applications. As opposed to existing capture methods, which typically require expensive setups, video sequences, or manual intervention, our solution can capture at scale, is agnostic to the optical appearance of the textile, and facilitates fabric arrangement by non-expert operators. To this end, we propose a sim-to-real strategy to train a learning-based framework that can take as input one or multiple images and outputs a full set of mechanical parameters. Thanks to carefully designed data augmentation and transfer learning protocols, our solution generalizes to real images despite being trained only on synthetic data, hence closing the sim-to-real loop. Key in our work is to demonstrate that evaluating the regression accuracy based on the similarity at parameter space leads to inaccurate distances that do not match the human perception. To overcome this, we propose a novel metric for fabric drape similarity that operates on images instead of the parameter space, allowing us to evaluate our estimation within the context of a similarity rank. We show that our metric correlates with human judgments about the perception of drape similarity, and that our model predictions produce perceptually accurate results compared to the ground truth parameters. The contributions in this chapter led to the following publication[1]:

[1] Additional publication details and results are included on the project website.

"How Will It Drape Like? Capturing Fabric Mechanics From Depth Images"

Carlos Rodriguez-Pardo, Melania Prieto-Martin, Dan Casas, Elena Garces

Computer Graphics Forum (Proc. Eurographics 2023) (2023)

6.1 Introduction

Creating accurate digital representations of real-world materials, or *Digital Twins*, is crucial for enabling realistic 3D visualizations suitable for interactive design and predictive engineering. Some industries, like fashion or textile manufacturing, further require these methods to work at a scale to cope with the fast pace of the current production workflows. However, digitizing cloth is challenging due to the high variability and type of fabric samples, where the fabric composition, the microstructure, or the finishing play crucial roles in the perceived appearance.

While casual systems which obtain optical appearance have long been a focus of research, comparably less attention has been paid to estimating *mechanical* properties. Indeed, capturing and simulating the mechanical behavior of cloth is challenging due to the complex interplay between the internal and external forces occurring in this type of physical system, which is highly sensitive to the environmental conditions. Nevertheless, with the current need to create virtual copies instantly, a casual setup able to produce automatic and accurate estimates–beyond having a set of *presets* from which to manually choose the closest one–of mechanical parameters could prove very valuable.

Previous methods are impractical for scalable and customizable workflows. Accurate fabric parameter acquisition systems require specialized and expensive devices [Kaw80; Min95; Cla+90], which are often slow and need skilled operators. Existing casual capture setups use input video sequences [Bha+03; Bou+13; YLL17] or, even if they take a single image, might require manual user input [JC20]. In this chapter, we present a casual capture system that only requires taking two depth images of the textile posed in a static drape. Our capture setup does not require complex calibration, can be easily manipulated by non-expert operators, and is agnostic to the optical properties of the fabric

thanks to leveraging depth images instead of RGB data. Figure 6.2 illustrates our capture setup. It involves capturing the fabric with a depth camera in two relaxed positions: the *hanging* scene, whuch conveys the drape when no force other than gravity is applied to it, and the *stretch* scene, which provides cues on the stretching properties of the fabric.

We propose a learning-based method to instantly return the mechanical parameters given static depth images and the fabric density as input. Our method relies on a sim-to-real strategy [Zhu+21], leveraging transfer learning and building on a dataset of physical fabrics digitalized with precision equipment. Our model is trained solely with synthetic data, and thanks to carefully designed policies of data augmentation and neural features, it can gener-alize to real-world scenarios. Under the hood, our approach leverages a custom architecture that enables a flexible design, that can take one or multiple images as input, enhancing its performance when more data is available. We perform an extensive evaluation by means of ablation studies and by measuring aggregated neural network saliency maps, which show that some scenes are more informative than others for predicting the mechanical properties of the fabric. Furthermore, we demonstrate the performance of our model on real-world captured samples, showcasing our system's generalization capabilities.

Figure 6.2: Capture setup, RGB (top), and depth (bottom) images for *hanging* (left) and *stretch* (right) scenes in rest position. Each scene conveys a different mechanical appearance of the fabric: *hanging* exhibits the overall drape; *stretch* exhibits an extra diagonal tension, which is key to understand the stretching properties.

Key to our work is to demonstrate that evaluating the prediction accuracy of mechanical properties using typical error metrics, such as the Mean Absolute Error (MAE) on the parameter space, leads to inaccurate distances that do not

match the human perception. We identify that such mismatches occur due to two factors: first, the parameter space is not bijective–i.e., different sets of parameters might convey the same drape–; and second, a numerical error in a parameter does not necessarily correlate with what we perceive as an error. To address this shortcoming, common in all existing works, and inspired by previous work on similarity metrics for material appearance [Lag+19], natural images [Zha+18] or illustration [Gar+14], we propose an image-based metric that measures differences on the mechanical behavior of textiles taking into account the overall drape. We validate that our metric agrees with human perception, and it can be used to sort materials by drape similarity with respect to a reference fabric.

Using our similarity metric, we finally validate that the estimations of our method correlate with human judgments about drape similarity, and that our model predictions produce perceptually accurate results compared to the ground truth parameters. All in all, our approach makes an important step towards solving sim-to-real problems for mechanical estimation since it shows that simulated cloth using inferred parameters maximizes the similarity with respect to real-world target fabrics.

6.2 Related Work

6.2.1 Parameter Estimation Methods

Estimating the mechanical properties of real fabric samples is a highly challenging problem for several reasons: the number of uncontrollable extrinsic factors (e.g., wind forces, initial state, collisions, etc.) which affect the predictiveness of the physical simulation; the lack of a standard deformation model and parameter spaces; and the use of computation-intensive simulation methods. Accordingly, a wide range of strategies exist, aimed at overcoming these challenges.

Measurement Devices. It is common to combine optimization techniques with the output of testing devices to find the optimal set of parameters which best explain the observations [Mag+07; SB08; VMF09; WOR11; Mig+12; CTT17]. Existing technologies of this type are diverse and, as discussed by Kuijpers *et al.* [KLG20], lack of a clear standard. The Kawabata Evaluation System (KES) [Kaw80] is perhaps one of the most well known, measuring 16 coefficients including bending, shearing, and tensile among others. Despite its precision, this method was not widely adopted by the industry due to its lengthy processes and the need for expensive equipment. Consequently, several other methods tried to simplify and unify the methodology with partial success

according to some studies [LM08; Pow13]: the Fabric Assurance by Simple Testing (FAST) [Min95], the Fabric Touch Tester (FTT), the CLO Fabric Kit 2.0, the Fabric Analyser by Browzwear (FAB), the Optitex Mark 10, and the cantilever principle [Cla+90].

Reconstruction-Optimization Methods. Another set of techniques jointly tackles the reconstruction and parameter optimization problems. By taking as input data from arbitrary real simulations (e.g., the cloth deforming on an avatar [Yan+18]), they iteratively reconstruct and simulate the scene which better explains the observation. Bhat *et al.* [Bha+03] takes as input a video sequence of the cloth and uses simulated annealing to optimize the parameters by measuring its folds. Yang *et al.* [YL15] use multi-view stereo reconstruction to initialize the 3D shape. Runia *et al.* [Run+20] also introduces simulation steps to explain observed phenomena of cloth in the wind. They rely on similarity metrics computed on deep latent spaces to supervise the optimization of the parameters. These methods require simulation steps embedded into the fitting processes, making them computationally expensive due to the high dimensionality of the parameter spaces. Recent differentiable simulation techniques [LLK19; JBH19; Hu+19; Mur+20; Li+22] have proven to be efficient ways to reduce the fitting burden by enabling the computation of gradients with respect to these parameters within latent spaces of neural networks, taking into account dynamics, self-collisions, and contacts.

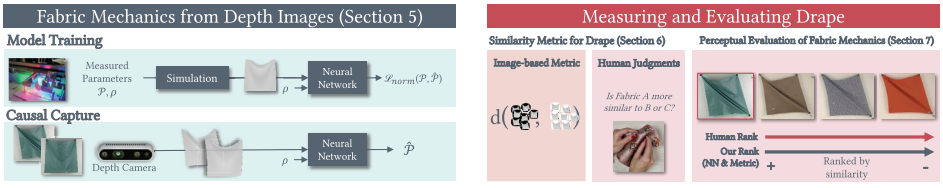
Data-Driven and Regression Methods. The third set of methods avoids reconstructing the original 3D scene by working on an estimated feature space and leveraging previously simulated data and machine learning techniques. Our approach falls into this category. Taking videos as input has been explored by Bouman *et al.* [Bou+13] to recover stiffness and area weight using a descriptor of the image based on PCA and optical flow, and later by Yang *et al.* [YLL17], who leverage neural networks to extract image features used for regression. Davis *et al.* [Dav+15] estimated the same simulation parameters by exploiting imperceptible vibrations in high-speed video recordings. Bi *et al.* [Bi+18] further evaluated that humans also need fabric motion to understand its stiffness. Friction coefficients have been estimated using reflectance values [ZDN16] or dynamic videos of cloth sliding through a surface [Ras+20] Instead of regressing the parameters, Huber *et al.* [HEW17] find the most similar cloth in a database using motion descriptors. A different approach only using a single image of the *Cusick drape* was followed by Ju *et al.* [JC20], but it requires a 360° scan to reconstruct the target cloth, and a manually fitted Bezier curve to obtain the feature vector. In contrast, we just require a depth map that can be captured easily. Concurrent work [Fen+22] uses multiple-view depth images as input to a trained regressor.

Our approach is inspired by these ideas; however, we do not require optimization–providing instant estimation of the parameters– and leverage neural features to understand and model fabric behavior in a semi-controlled setup. We demonstrate that our approach works with two images as input to predict bending and stretching coefficients without requiring a full video of the piece of fabric.

6.2.2 Pre-Trained Models and Transfer Learning

Deep learning models typically require vast amounts of data for generalizing to unseen examples. When this amount of data is not possible to acquire, *transfer learning* techniques help by re-using model parameters trained on a related task [KSL19; Rag+19; Bom+21]. These techniques include: *Fine-tuning* the weights of a pre-trained classification model [WKW16; RD17; Che+18; Wan+18; RJ19; Guo+19; Kol+20]. Pre-training an image descriptor model on contrastive or self-supervised learning tasks, and uses the activations of its last layer as input to the downstream task (*Linear Probing*) [Che+20b; Rad+21; CXH21; He+22; Kum+22]. For domain adaptation problems, it is common to *adapt* the internal representations of pre-trained CNNs so as to efficiency [RBV17; RBV18; RB19; Pha+20; LLB22]. Inspired by these approaches, we design a model that leverages fine-tuning of a pre-trained image CNN classifier as a feature extractor, capable of processing depth images, and extending it to account for additional input variables, and handling multiple images at the same time during inference.

Figure 6.3: An overview of the main components of our method. We propose a system to estimate fabric mechanics using depth images of *hanging* and *stretch* scenes as input. To validate the error of our estimations perceptually –accounting for the global drape–, we propose an image-based drape similarity metric which we validate with human judgments and can be used to sort fabrics by similarity. We show through several metrics that the estimations provided by our method using our similarity metric agree with those given by humans.



6.2.3 Similarity Metrics

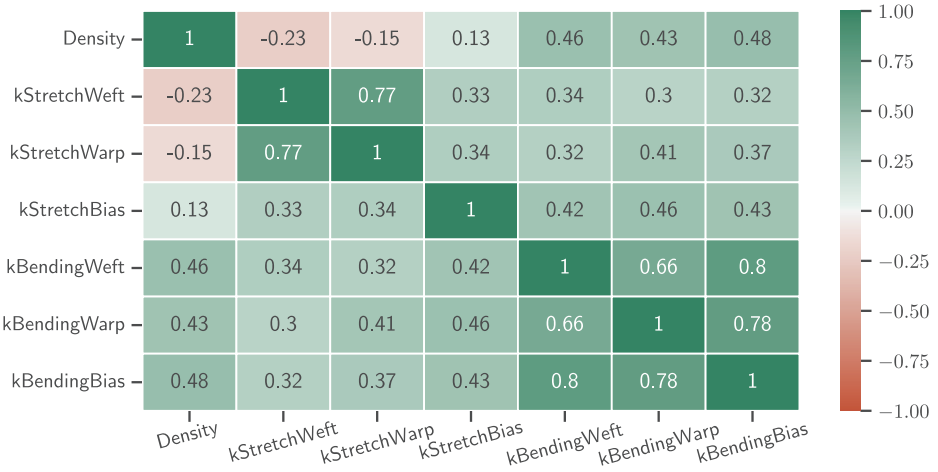
Full-Reference Image Quality Assessment (IQA) aims to provide a single score that measures the amount of distortion between two images. Traditionally, these metrics leveraged low-level image statistics. *PSNR* is commonly used for measuring image degradation, but correlates poorly with human perception [Zha+18]. More sophisticated alternatives have been developed, including *SSIM*, *mSSIM* [Wan+04], and others [Zha+11; Naf+16; ZSL14; ZL12; Rei+18]. Algorithms based on latent spaces of CNNs [GEB16a] have been extended to better approximate human perception, for example, by training on a large pool of human evaluations *LPIPS* [Zha+18], or by other means [Din+20; Pra+18].

Besides, similarity metrics that measure abstract or complex concepts like style have been proposed for 3D furniture [LKS15], illustration [Gar+14], icons [LGG19], product design [LKS15], or material appearance [Lag+19]. Unlike ours, these metrics require to be trained with human ratings, thus incurring a considerable cost to collect such information via user studies. Instead, our metric does not require specific training, leveraging an off-the-shelf image-based metric. Despite this, we show that our metric correlates with human judgments on the perception of fabric drape similarity, and that can be used to evaluate the overall drape.

6.3 Overview

Figure 6.3 presents an overview of our work. First, in Section 6.5, we introduce our novel solution to infer fabric mechanics directly from depth images. As input, our approach only requires static depth images in two specific configurations, shown in Figure 6.2, as well as the fabric density which can be easily obtained with conventional equipment. In Section 6.4 we describe the datasets of *synthetic* and *real* samples, with mechanical ground truth parameters, used train and evaluate our regressor.

Figure 6.4: Spearman correlation matrix between parameters of our synthetic dataset.

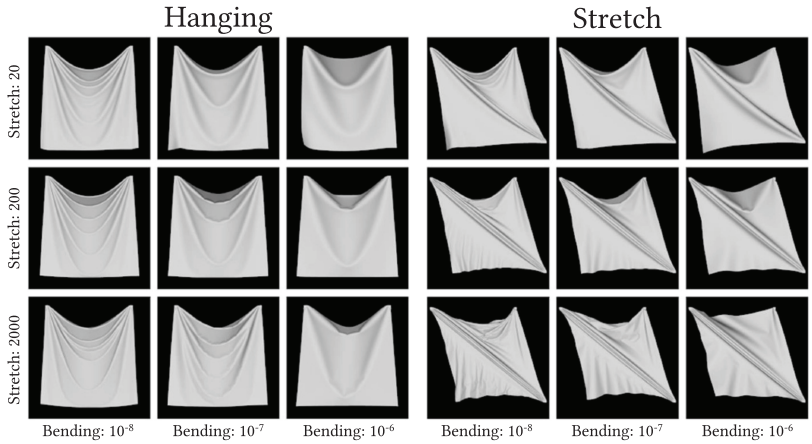


The quantitative evaluation suggests that our method is able to estimate the mechanics within a certain error. However, since a direct interpretation of that error is not human-friendly, we propose a method to evaluate the overall drape in the context of a real scene. In Section 6.6, we introduce our image-based similarity metric for drape, which takes as input renders of the chosen scenes and provides a relative value that is useful to compare the drape of different fabrics. With a user study, explained in Section 6.7.1, we validate that our metric agrees with human preferences on the global perception of fabric drape. Also, in Section 6.7.2, we evaluate our capture method using our drape similarity metric and compare it with human judgments. We effectively validate that our estimations agree with human assessments and provide several qualitative examples in Section 6.7.3.

6.4 Datasets

We develop two different datasets of depth images, which we use at different steps of the pipeline to train and evaluate our models: a *synthetic* dataset, generated using physics-based cloth simulation; and *real* dataset, generated using images of real fabric samples. In both datasets, a sample consists of a depth image of a fabric simulated in a scene for which we have the corresponding mechanical parameters, namely: bending [Gri+03] and stretch [VMF09] in the warp, weft, and bias directions and the fabric density, $\{$*kStretchWarp*, *kStretchWeft*, *kStretchBias*, *kBendinghWarp*, *kBendingWeft*, *kBendingBias*, $\rho\} \in \mathbb{R}^7$. We support two different static configurations for the scenes: *hanging*, which exhibits the overall drape; and *stretch*, which exhibits diagonal tension and is key to understanding the stretching properties. Figure 6.2 depicts examples of each configuration.

Figure 6.5: Sweep of simulation parameters for hanging and stretch scenes. For *kBending*: warp, weft, and bias have the same value, while for *kStretch*, bias changes as 100, 144, 1000.



Simulated Dataset. To train our model, we generate a synthetic dataset by simulating fabrics in a virtual scenario, replicating the *hanging* and *stretch* configurations. We use a standard simulator, similar to ARCSim [NSO12], with a quadratic strain, a linear strain/stress relationship, and standard definitions for bending and stretch [Gri+03; VMF09]. To model highly anisotropic fabrics, we use three parameters for warp, weft, and bias. Note that the model used for stretch already has some nonlinear behavior (quadratic strain), but more parameters (one per direction) are required to control the nonlinearity of the forces. Thickness is not included as it is implicitly accounted for in the other parameters. Contrarily, density is required as the simulator is dynamic. In a static one, it could be dropped after normalizing the other parameters.

After simulation, we render the resulting mesh (discretized at 5mm/edge) with a white Lambertian material and extract the depth buffer. To ensure that our synthetic dataset covers a wide range of materials, we densely sample the parameter space using a distribution of common mechanical parameters of real-world fabrics. Figure 6.5 shows a sweep of parameters showcasing the variability of resulting drapes in the hanging and stretch scenes. To better understand potential relationships between the parameters, we compute the Spearman correlation *r*, shown in Figure 6.4, where we observe the higher correlation between the three kStretch coefficients and some correlation between the kBendingBias and the density.

Real Dataset. To evaluate our model, we test it with real data from images captured by the Intel RealSense SR300. To this end, we casually hang 50 × 50 cm fabric samples using several magnets into a metallic panel, which

requires little to no expertise and can be done very fast. Pin location does not need to be centimeter-accurate. Because we use depth images, no special lighting is necessary. We measure fabric area density by weighing a 10 × 10 cm sample and dividing by its area. See Figure 6.2 for a visualization of our capture setup and the accompanying video for an illustration of the process. We capture ten fabrics of diverse compositions and structures, for which we have ground truth mechanical parameters previously obtained with specific equipment and methods [Spe+22].

6.4.1 Data Augmentation

To make our model robust to potential noise and scenario variations common in uncontrolled capture setups (e.g., slightly different camera viewpoint or fabric configuration), we apply several data augmentation strategies to our *synthetic* dataset.

Simulation-Space Data Augmentation. To enforce robustness to different camera view-points, simulated meshes are rendered within a range of different inclinations with respect to the vertical plane. This range covers ±5 degrees from the rest position orientation, creating 11 depth maps per material and scene.

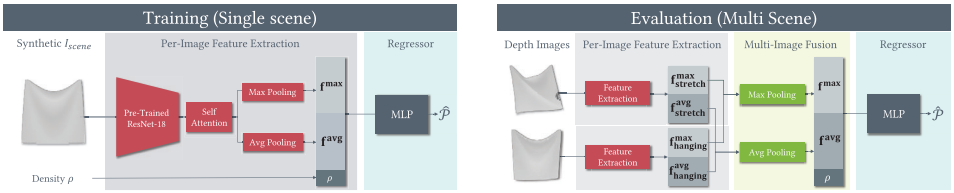
Image-Space Data Augmentation. Real images are largely different from synthetic images, due to distortions, noise, perspective changes, unknown illumination, lens and sensor characteristics, blurs, etc. To enforce the robustness of our models to those distortions, we design an extensive image data augmentation policy consisting of random individual deformations, performed in a particular order. These not only include random noise, blurs, perspective changes and rescales, but also more complex policies such as thin-plate deformations, posterization and erasing. This data augmentation policy bridges the gap between synthetic renders and real depth images, which are typically noisier.

6.5 Fabric Mechanics from Depth Images

In this section, we present our learning-based approach to estimate fabric mechanical param-eters $\hat{\mathrm{P}}$ from depth images. Given a set of depth images, $I=\{I^{\text{hanging}}_{i},I^{\text{stretch}}_{i}\}_{i}\geq 1$, which depicts the *hanging* and *stretch* scenes, we train a model M which maps $\mathbf{I}$, along with the material *density* ρ, to mechanical parameters: $\mathrm{M}(\mathbf{I},\rho)=\hat{\mathrm{P}}\in\mathbb{R}^{6}$. To this end, at train time we learn to extract relevant features from depth images, which are then fed into a regressor to learn to

predict mechanical features. Importantly, our architecture enables to input sets of images at test time by fusing their respective features. Figure 6.6 illustrates the train and test pipelines.

Figure 6.6: Diagram of our training and evaluation pipelines. For **training**, we use a single image of the material, along with its density. The image is processed by our *Feature Extractor*, followed by an MLP, which computes the parameter estimation $\hat{\mathrm{P}}$. For **evaluation**, we process each available image with our feature extractor and use a *fusion* operator before feeding it to the trained regressor. We use the same Feature Extractor and MLP for both scenes.



6.5.1 Neural Network Architecture

Feature Extractor The first part of the model is a *feature extractor* F, which receives as input a single depth image $\mathrm{I}_{\mathrm{sc}} \in \mathrm{I}$ and outputs a feature vector $\mathrm{f}_{\mathrm{sc}} = \mathrm{F}(I_{\mathrm{sc}})$ that describes it. This feature extractor is composed of three different components, shown in Figure 6.6. First, the image is processed by a Convolutional Neural Network (CNN) that outputs a dense latent representation. We use a ResNet-18 [He+16] pre-trained on ImageNet [Den+09], which we finetune with our data. To reduce the gap between real and synthetic images, we equalize the image before feeding it to the feature extractor, during both training and evaluation. Then, the output of this CNN is passed through a *Self-Attention* [Zha+19b] module, which helps the model learn non-local dependencies, enlarging the model receptive field, so it accounts for distant information in the input images. Self-Attention mechanisms were originally designed for language models [Vas+17] but have recently demonstrated significant efficacy for computer vision tasks [Ram+19; Zha+19b; ZJK20; Han+22b]. We add a single Self-Attention layer, as they are expensive to train and evaluate.

Finally, we perform pooling operations to transform the output of the Self-Attention layer to a feature vector, f_{sc}, of a fixed size. In addition to the commonly used max-pooling [SZ15], we further concatenate it with the output of average-pooling, which has been shown to improve the performance of attention modules [Zho+16; HSS18; Woo+18]. The feature vector f_{sc} is thus

Table 6.1: Ablation study of the neural architecture and data augmentation. From left to right, we build upon our baseline and progressively add: simulation-space data augmentation, image-space data augmentation, pre-training, self-attention, and average-pooling. On both MAE (ℓ_1) and correlation (s) metrics, we observe increased performance on the validation set in every added component. Using a pre-trained network for feature extraction yields the largest gains. We use a color code to highlight **best** and **worst** cases.

Metric	Parameter	Baseline	Data Augmentation			Architecture	
			w/ sim	w/ image	w/ pre-Train	w/ attention	w/ pooling
$\ell_1\downarrow$	kStretchWeft	0.122	0.118	0.112	0.087	0.071	0.071
	kStretchWarp	0.109	0.101	0.102	0.074	0.066	0.061
	kStretchBias	0.050	0.043	0.046	0.043	0.042	0.038
	Avg. Stretch	0.094	0.087	0.087	0.068	0.060	0.057
	kBendingWeft	0.095	0.093	0.091	0.084	0.0.82	0.072
	kBendingWarp	0.124	0.119	0.112	0.110	0.101	0.094
	kBendingBias	0.086	0.081	0.086	0.081	0.068	0.060
	Avg. Bending	0.102	0.098	0.097	0.092	0.084	0.075
r↑	kStretchWeft	0.445	0.475	0.453	0.643	0.788	0.798
	kStretchWarp	0.543	0.503	0.510	0.645	0.715	0.771
	kStretchBias	0.477	0.508	0.608	0.701	0.733	0.781
	Avg. Stretch	0.488	0.495	0.523	0.660	0.745	0.783
	kBendingWeft	0.611	0.684	0.781	0.783	0.798	0.863
	kBendingWarp	0.614	0.654	0.747	0.772	0.806	0.921
	kBendingBias	0.624	0.828	0.837	0.872	0.893	0.942
	Avg. Bending	0.616	0.722	0.788	0.809	0.832	0.909

a concatenation of max-pooled features and average-pooled features: $\mathrm{f}_{\mathrm{sc}} = \left\{ \mathrm{f}_{\mathrm{sc}}^{max} \oplus \mathrm{f}_{\mathrm{sc}}^{avg} \right\}$.

Fusion Our design allows us to combine features f_{sc} from more than one scene into a single feature vector **f**. For every image in I, we compute f_{sc}. We then *fuse* those feature vectors into a single vector by performing pooling across f. Similarly to f_{sc}, f is compos types of features: $f = \left\{ \max_{sc} \left\{ f_{sc}^{\max} \right\} \oplus \left\{ avg_{sc} \left\{ f_{sc}^{avg} \right\} \right\} \right\}$ The max-pooled features are fused using the maximum value across scenes, while the average-pooled features are fused using the mean across scenes. We illustrate this procedure in Figure 6.6.

Parameter Regressor Our last component is a fully-connected Multi-Layer Perceptron (MLP), which takes as input the feature vector f and the material density, ρ, and outputs the simulation parameters $\hat{\mathrm{P}}$. As our loss function, we compare the real parameters P with the model estimations $\mathrm{M}(\mathrm{I},\rho)=\hat{\mathrm{P}}$ using an ℓ_2 norm.

6.5.2 Quantitative Evaluation

In this section, we quantitatively evaluate the performance of the method for estimating the mechanical parameters. We validate the design choices of the model and evaluate our results depending on the type of input used.

Ablation Study of the Model Design

We aim to understand the effective contribution of the data augmentation strategy, and the network architecture design. For these experiments, we randomly split the synthetic data in 90% for training and 10% for validation, using the same split for every experiment. Results are shown in Table 6.1. As our baseline, we use a simple model without *self-attention*, without *average-pooling* features, where the CNN backbone is randomly initialized, and without data augmentation. From this baseline, we progressively add different components and measure the performance of the validation data using Mean Absolute Error (MAE) and Spearman correlation (r). The parameters are normalized using the minimum and maximum values of the training set.

Given the training configuration with all the data augmentation –which provides a small increase in performance most likely because the validation dataset is synthetic data– we evaluate the neural architecture.

Using a CNN backbone pre-trained on ImageNet [Den+09] instead of a randomly initialized one, we observe a significant increase in model performance across every parameter and metric. Training a feature extractor that receives

images from both scenes at the same time would not allow us to leverage pre-training, which would negatively impact generalization. Then, we add a *self-attention* [Zha+19b] layer after the CNN backbone, which allows the model to integrate information that is present in distant areas of the images. Interestingly, this module significantly helps predict the *kStretch* parameters while having a more minor influence on the *kBending*. Finally, adding *average-pooling* in addition to the commonly used *max-pooling* has a highly positive impact on the error rates. We use this last configuration with all the components for all the results shown in the chapter. It is worth noting that the *kBendingBias* is easily predicted by the model, even in its most basic configuration. This is likely because this parameter correlates most strongly with the density of the material (shown in Figure 6.4), so the model can leverage this information for the predictions.

Evaluation of Input Influence

The design of our method supports taking as input one or multiple images. In this experiment, presented in Table 6.2, we evaluate the error testing different configurations of the input. Note that we train a different model for each configuration. In every case, using the *density* as input helps the model to generalize. This is particularly relevant for *kBending* parameters, for which the *density* alone provides more information than depth images. The *stretch* scene is typically more informative than the *bending* one, as both MAE and correlations are usually better when it is provided. When using both scenes simultaneously, the model provides moreaccurate estimations than any of the scenes individually, showing that the two scenes provide complementary information.

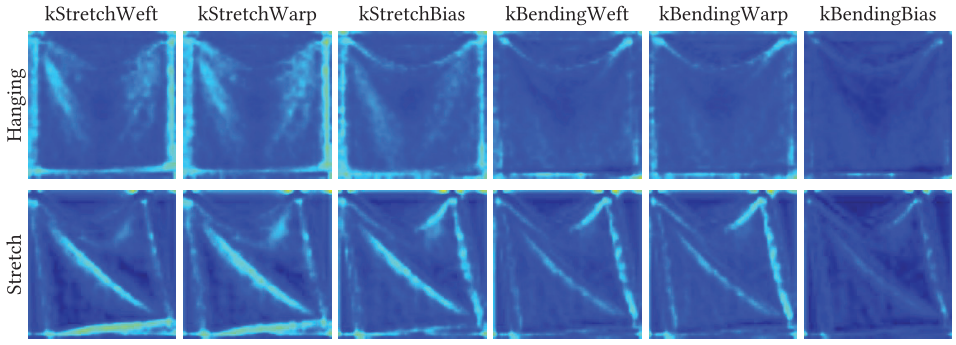
Table 6.2: Results for real depth images varying the input. From left to right, the input is: only density, only depth images, and both density and depth. The best results are obtained using every data source as input. For *kBending*, the *density* alone provides more information than only depth images. We use a color code to highlight **best** and **worst** cases.

Metric	Parameter	Density	OnlyDepth			Density & Depth		
			Stretch	Hanging	Both	Stretch	Hanging	Both
$\ell_1 \downarrow$	kStretchWeft	0.113	0.102	0.107	0.062	0.054	0.056	0.051
	kStretchWarp	0.091	0.067	0.081	0.056	0.055	0.059	0.052
	kStretchBias	0.034	0.039	0.054	0.034	0.033	0.036	0.031
	Mean Stretch	0.079	0.069	0.081	0.051	0.047	0.050	0.045
	kBendingWeft	0.142	0.233	0.213	0.145	0.128	0.139	0.125
	kBendingWarp	0.126	0.184	0.126	0.074	0.037	0.063	0.035
	kBendingBias	0.094	0.166	0.081	0.054	0.055	0.069	0.046
	Mean Bending	0.121	0.194	0.140	0.091	0.073	0.090	0.069
r $\uparrow$	kStretchWeft	0.184	0.407	0.403	0.418	0.712	0.469	0.728
	kStretchWarp	0.002	0.502	0.407	0.503	0.520	0.433	0.533
	kStretchBias	0.289	-0.141	-0.068	-0.007	0.367	0.383	0.550
	Mean Stretch	0.158	0.256	0.247	0.305	0.533	0.428	0.604
	kBendingWeft	0.483	0.052	0.267	0.625	0.673	0.683	0.717
	kBendingWarp	0.317	0.048	0.156	0.407	0.467	0.433	0.533
	kBendingBias	0.357	-0.044	0.108	0.250	0.333	0.417	0.546
	Mean Bending	0.386	0.019	0.177	0.427	0.491	0.511	0.599

Neural Saliency Maps

We aim to understand which parts of the scenes are most relevant for the model when making its predictions. To do so, we create saliency maps using *Axion-Based Class-Activation Mappings* [Fu+20], which averages the activations of the deep layers weighted by their importance with respect to each target parameter, and aggregate them over our real dataset. In Figure 6.7 we observe that each scene provides the model with different cues. For *kStretch* parameters, the model is sensitive to the central wrinkle of the *stretch* scene and the central fold of the *hanging*. For *kBending* parameters, the model is sensitive to areas in the borders of the fabric where small but noticeable wrinkles are present, in both scenes. As in other experiments, we observe that the model can find more relevant features on the *stretch* scene.

Figure 6.7: Saliency maps [Fu+20] aggregated per parameter. The model relies on the central areas of the fabric samples for predicting the stretch parameters. For bending, it is most sensitive to areas on the borders of the samples.



6.6 A Similarity Metric for Drape

The regressor introduced in Section 6.5 allows us to infer the mechanical parameters of a target fabric. However, since a direct interpretation of such parametric space is not human-friendly, it is very difficult to understand the residual errors shown in Tables 6.1 and 6.2. Do the regressed parameters produce a drape similar to the target image? Notice that, since the mechanical parameter space is non-orthogonal, small parameter changes may produce unexpected deformations. Therefore, we hypothesize that a *perceptual* similarity metric for drape is needed to interpret our quantitative results. We describe this metric next.

6.6.1 Image-based Similarity of Drape

Motivated by our hypothesis that the *hanging* and *stretch* scenes are sufficient to convey the fabric mechanics, we propose an image-based similarity

metric using renders of such scenes. Let $\mathrm{P} \subset \mathbb{R}_7$ be the parameter space of our simulator, $\mathrm{P}_a \in \mathrm{P}$ and $\mathrm{P}_b \in \mathrm{P}$ STIX Two Math two different parameter sets, and $a \sim \mathrm{R}(\mathrm{P}_a, \text{scene})$ and $b \sim \mathrm{R}(\mathrm{P}_b, \text{scene})$ two rendered simulations obtained for a certain scene configuration. We define a distance metric for a particular scene as:

$$d_{\text{scene}}\left(P_a, P_b\right) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \mathrm{IM}\left(a_i, b_j\right)}{N^2} \tag{6.1}$$

where IM is an image-space distance metric, and N is the number of different simulations we run. Since real cloth is very sensitive to parameters such as initial state or initial shape, in order to learn a metric that is robust to real-world conditions, we perturb the initial state and boundary conditions in a set of simulations using random jittering to the initial forces. We empirically found that averaging over multiple simulations for the same set of (P_a, P_b) gives us a more informative metric.

Furthermore, we take into account both *hanging* and *stretch* scenes, hence our final metric is defined by averaging their distances across both scenarios, resulting in our final metric:

$$d\left(P_a, P_b\right) = \frac{d_{\text{hanging}}\left(P_a, P_b\right) = d_{\text{stretch}}\left(P_a, P_b\right)}{2} \tag{6.2}$$

Note that we propose a similarity metric, which, as opposed to real distance metrics, does not necessarily have to meet the metric axioms [TK74]: it can produce asymmetric values, violate the triangle inequality, and does not need to define what the identity is. According to Equation 6.1, the distance of one fabric with itself is not necessarily zero; it just needs to satisfy a minimum requirement where the distance of every material with itself should be smaller than the distance of any material with any other material,

$$d\left(P_a, P_a\right) < d\left(P_a, P_a\right), \forall P_a \neq P_b \tag{6.3}$$

In order to remove the possible influence of optical properties, scene illumination, and camera parameters, we use the same scene configuration for every render, with grayscale albedo and a Lambertian BRDF.

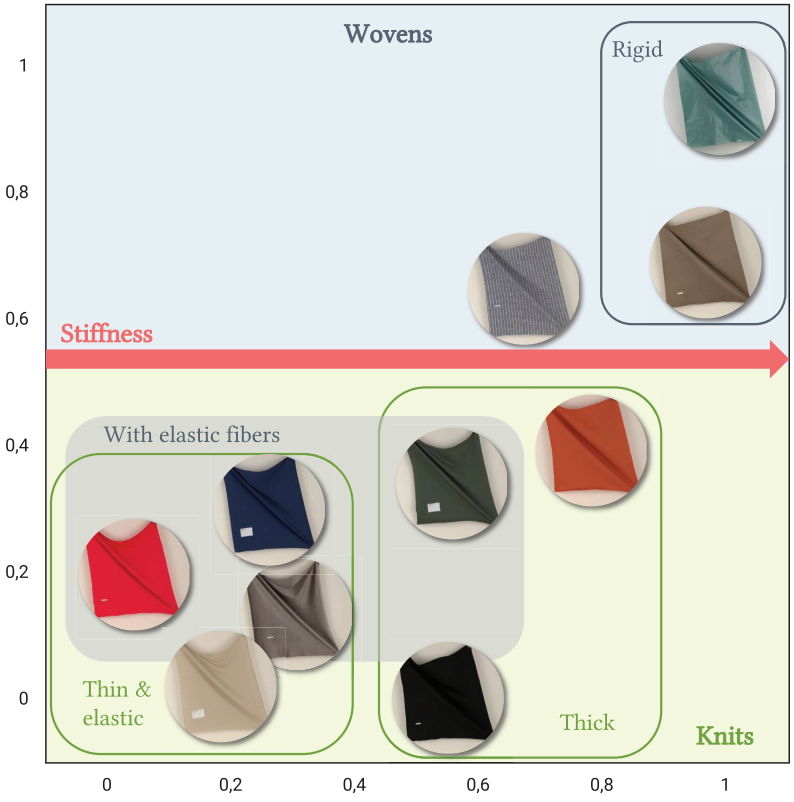
For IM we use LPIPS [Zha+18] which we empirically found to perform better than other alternative image metrics. We find that metrics based on pre-trained neural networks work better than lower-level alternatives, while content-aware distances are more powerful for this purpose than style-aware metrics. This suggests that the size, position, and shape of the wrinkles and deformations of the fabrics are important factors that explain differences between materials. We empirically found that $N = 5$ simulations are typically enough, as more samples provide very marginal improvements.

See the supplementary material of this chapter for more details about the proposed metric.

6.7 Evaluation

We propose to evaluate our method from Section 6.5 and the metric from Section 6.6 by comparing our estimations with human ratings (recall that we provide quantitative errors per parameter in Section 6.5.2). To this end, in Section 6.7.1, we first collect a large number of ground truth human judgments about the similarity of triplets of fabrics. Then, in Section 6.7.2, we demonstrate that our image-based metric using ground truth parameters, as well as the estimated ones, encode the same preferences. Finally, in Section 6.7.3, we show qualitative

Figure 6.8: Human Judgments 2D tSTE embedding [Tam+11] computed from human perceptual judgments about real fabric similarity (i.e., ground truth). We observe interesting patterns: woven and knits are separated; elastic materials are clustered; thick and thin materials are separated. Neither axes directly correspond to any material property, instead they emerge from the embedding.



comparisons and demonstrate the usefulness of our approach in a downstream task consisting of 'search by similarity'.

6.7.1 Human Judgment Perceptual Similarity of Drape

We then use ten samples from our real dataset with known ground truth mechanical parameters and setup the user study as follows. Participants are presented with a triplet of fabrics and, using one fabric as a reference, they are asked which of the two remaining fabrics is most similar [Zha+18; Gar+14] to the reference fabric. Participants are encouraged to manipulate the samples and focus only on the mechanical similarity and overall drape, and to ignore properties like material reflectance. Each participant rated 20 triplets that were pseudo-randomly sampled, ensuring that at least each of the ten test fabrics is used twice as a reference. Given the same triplet, we observe an average of 86.68% agreement between our participants across all experiments and materials, suggesting that there is a perceptual understanding of fabrics mechanics that humans share. We did not observe any significant differences in agreement depending on the volunteer demographics or level of expertise in fabric handling or simulation.

Leveraging the user study described above, we can compute an embedding that captures the relative distance between real materials according to human perception (i.e., ground truth perceptual similarity). Figure 6.8 depicts such embedding in 2D, computed using tSTE [Tam+11], which allows us to calculate perceptual distances between materials using the Euclidean norm. The embedding depicts many interesting patterns, including perfectly separated woven and knitted fabrics; elastic and thin fabrics are clustered together; thick and thin knits are well separated. These patterns suggest that fabric structure, composition, and density play an important, non-linear role in the overall perception of the mechanical properties of fabrics.

6.7.2 Image-based vs. Human Perceptual Similarity

We evaluate the agreement between humans and our estimations within the context of a similarity rank. For each fabric of our real dataset, we compute the distance to the rest of the materials using different metrics: 1) the Euclidean distances on the Human Judgments tSTE embedding shown Section 6.7.1; 2) the z-score distances in the parametric space of the mechanical simulation, P;3) the distance using our similarity metric for drape explained in Section 6.6. We compare ground truth parameters and estimated ones for the second and third cases. The summary of results is shown in Table 6.3, and the complete analysis is presented in the supplementary material.

First, we demonstrate that our similarity metric using ground truth parameters correlates with human judgments. Figure 6.9 (top) illustrates the outcome using Spearman correlation(r). We observe that the correlation for each fabric is higher than 0.8, with an average of 0.893, showcasing a strong correlation. These results suggest that our metric, which only takes images of the *hanging* and *stretch* scenes as input, can measure distances between materials as humans would. Then, as shown in Figure 6.9 (bottom), we use our estimated parameters instead of the ground truth, reaching an average correlation of 0.680. Even though this value is slightly smaller than the ground truth, it is still significant to conclude the estimations of our model agree with human judgments. Note that the materials with higher correlation are those lying on the extreme areas of the embedding obtained in Figure 6.8 that have very distinct characteristics. Likewise, we also compute the distance for each material using merely the parameter spaces of the simulation. As can be seen, using this space does not produce correlated outputs with human judgments, reaching correlations below 0.44 in any case tested.

Figure 6.9: Correlation between the ordering provided by the Human Judgments (x-axis) and our drape similarity metric with (top) the Ground Truth simulation parameters (y-axis), and (bottom) the estimations of our model. We plot z-scores instead of the raw distances to help visualization.

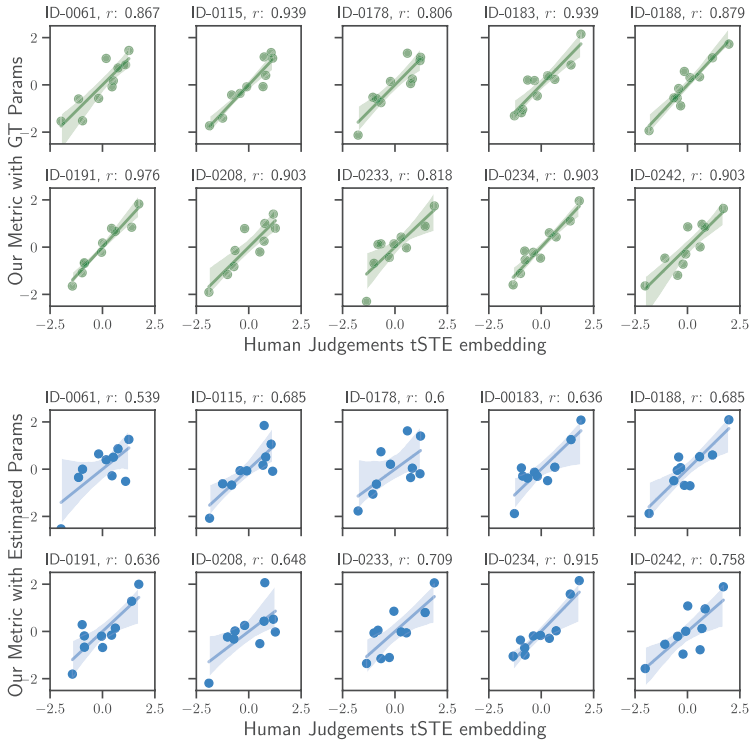


Table 6.3: Average (± std.) correlation with rankings obtained through the human judgments tSTE Embedding, depending on the parameter source (ground truth or predicted), and metric used to compute similarity (parameter distance or our drape similarity).

	Parameter Distance	Drape Similarity Metric
GT	0.431 ± 0.17	**0.893**±0.05
Estimated	0.377 ± 0.19	**0.680**±0.09

6.7.3 Qualitative Results

Figure 6.10 compares the simulations obtained with the parameters of our method with the ground truth parameters for a few examples. We can observe that our estimations are very close to the ground truth in this cases. Finally, our metric can be used to search between materials of similar drape. We illustrate this in Figure 6.11. As shown, a naive ranking using directly the parameter space does not provide any meaningful ordering. On the contrary, using our similarity metric, we obtain ranks that agree with those given by the human embedding, showcasing the potential of our automatic metric to explore fabric collections.

Limitations Even if our model provides accurate predictions, its estimations are not always truthful to the real materials. We illustrate this in Figure 6.12, where the model predicts fewer bends on the final drape than what the ground truth generates. According to our metric, this prediction is closer to thicker materials than to what humans perceive as most similar to the reference material.

6.8 Conclusions

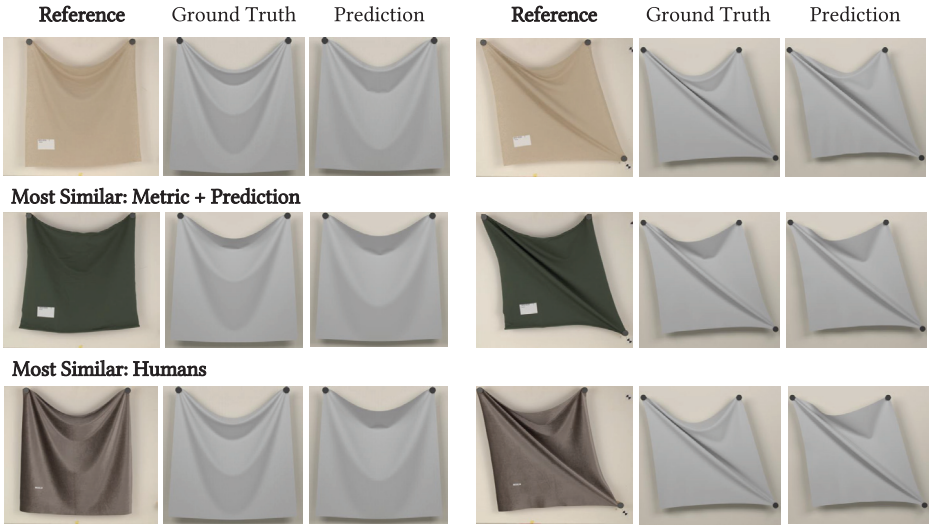
In this chapter, we have presented a casual method to estimate the mechanical parameters of fabrics from depth images of fabric samples placed at two specific configurations. We have validated our architecture and inputs numerically, proving that all the components of our method are necessary to provide accurate estimations. While our quantitative analysis helped us understand the importance of each component, we found that these errors are not interpretable, nor do they help us understand the overall appearance of the predicted drape. Therefore, we have presented the first metric, which, by purely working on the image space, can capture differences in fabric mechanics as humans do. We have used such metric for two purposes: first, to validate the accuracy of our estimated parameters perceptually, and second, to showcase a novel application of search by drape similarity.

Figure 6.10: A comparison between the simulations obtained through the ground truth parameters, and those obtained using the predictions of our model, from a representative set of fabrics of our test set. As shown, the estimations of the model yield similar drapes to those of their ground truth counterparts, which we also evaluate quantitatively.

Our work could be improved in several ways. Our neural network is trained using a pure regression loss. Training the network using differentiable simulation could improve training, and help generalization and error interpretation. We could incorporate our perceptual metric as a loss function. However, it requires multiple differentiable simulations and a deep feature extractor, which will result in a significant computational overhead. In addition, less expressive simulation engines may correlate less with human perception. Similarly, we would like to scale our training dataset and user study to handle more and more diverse samples, to cover a broader variety of fabric families. Interesting possible extensions would include taking as input RGB images instead of depth maps, training with real samples, incorporating symmetry consistency losses, or learning a similarity metric that can work by using captured images as input (instead of simulations). Finally, we hope our work might inspire future work in the long-standing problem of validating fabric mechanics in a way that is agnostic to the simulator parameters.

Figure 6.11: Search by drape similarity. The ordering provided by the parameter space (first row) does not match human judgments (second row), while the arrangement obtained by our metric matches humans with high consensus (third row).

Figure 6.12: A failure case of our method. Given the reference material (first row) as input, the model predicts fewer bends than the ground truth. According to our metric, this prediction is closer to a thicker material (middle row), than to what humans perceive as most similar to the reference fabric (bottom row).



6.A Additional Implementation Details

Training. We train our model using PyTorch [Pas+19] as our learning framework and Kornia [Rib+20] for image-processing operations, including normalization and data augmen-tation. The model is optimized using AdamW [LH17] for a total of 50 epochs, with a learning rate of α = 0.0003, which is halved every 10 iterations. All other optimizer hyperparameters followed the default AdamW configuration. We use mixed precision training [Mic+18], a batch size of 256, and a weight decay of 0.000002. The input images are of 180 × 180 pixels, and we use a Mean Squared Error (MSE, ℓ_2) as our loss function for our regression problem. Model size design and hyperparameter selection were conducted using Bayesian hyperparameter tuning [Bie20]. Training takes approximately one hour on an NVIDIA 2080 RTX GPU.

Network Design. Our feature extractor is a pre-trained ResNet-18 [He+16] (TorchVision pretrained weights checkpoint weights='IMAGENET1K V1'), followed by a Self-Attention layer [Zha+19b] with 64 attention heads. The output of this layer is max and average-pooled, and a final multi-layer perceptron (MLP) with 4 hidden layers with 512 hidden units each, receives the pooled features and the density value, and outputs the 6 predicted mechanical parameters. The MLP layers use ReLU [MHN13] non-linearities, followed by Layer Normalization [BKH16], and a Dropout [Sri+14] rate of 0.5. Both the feature extractor and the MLP are trained simultaneously.

Data Augmentation. We use the following policy, in order: First, we randomly rescale the input images using (0.5, 1.5) rescale ranges. We then randomly erase [Zho+20] parts of the input images (p = 0.2), randomly rotate them (p = 0.2, *angle* = ±10°), apply a random perspective change (p = 0.5, *scale* = 0.5), and a random thin plate spline warp (p = 0.2, *scale* = 0.5). We then crop the center area, with a resolution of 180 × 180 pixels. To those images, we randomly change their contrast and brightness, on the (0.5, 1.5) ranges. We then apply a final set of transformations, including image equalization, random horizontal flip (p = 0.5), random Gaussian noise (p = 0.5, μ = 0, σ = 0.02), random posterization ((p = 0.5)), random sharpening (p = 0.5), and random Gaussian blur ((p = 0.5), kernel size of 3 × 3). Please refer to the Kornia documentation [Rib+20] for specific implementation details of each of these transformations and the specific meaning of the mentioned parameters.

6.B Additional Results

In this section, we show the relative similarity rankings computed through the tSTE embedding of our user study, compared to the ground truth and predicted parameter distances, as well as the drape similarity computed using the ground truth and predicted parameters. We also provide additional details of our user study.

Figure 6.13: Relationship between the distance obtained by our user study, our distance metric, and the difference in each parameter. On the top row, we plot the difference between the value of a parameter against the average distance between fabrics we obtained through our user study. On the bottom, we show the same parameter distances, now against the distance predicted through our perceptual metric. As shown, no mechanical parameter dominates the distance predicted by either the user study or our metric. The perception of mechanical properties can be understood as a highly non-linear phenomenon in which many parameters interact in complex ways. We use z-scores to help visualization.

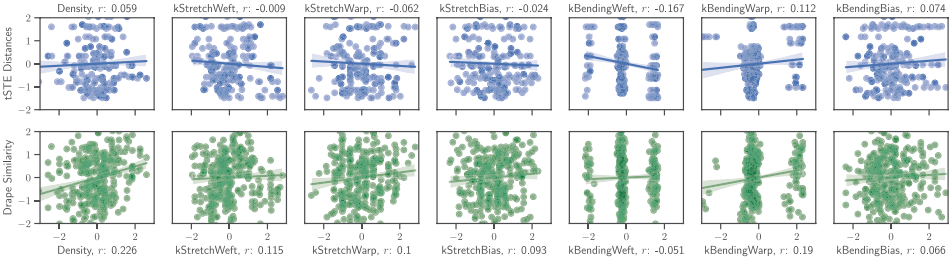
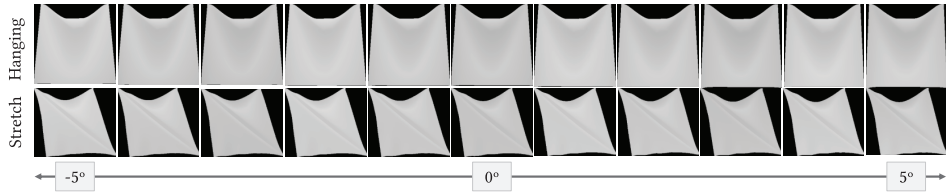


Figure 6.14: To recreate possible misalignments that may be present on the real images, we propose a *simulation-space* data augmentation policy, in which we simulate the drapes from different rest positions, on the [−5, 5] degree ranges. For every scene, we generate 11 different simulations uniformely in this range, effectively creating a dataset of images for a single material.

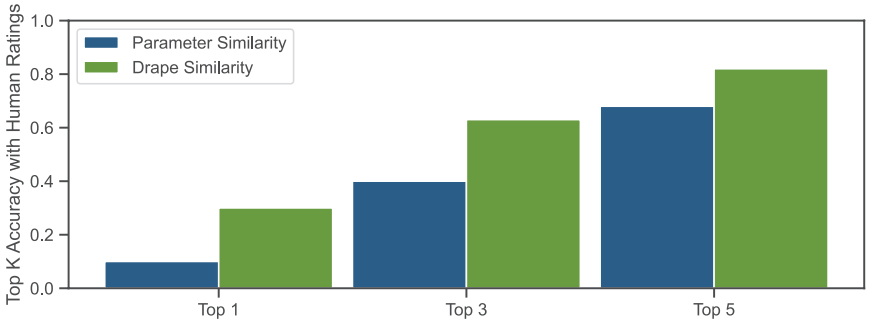


User Study: Procedure and Stimuli

Stimuli. We use ten physical fabric samples of our real dataset, which are representative of different types of fabric structures, thicknesses, densities,

and mechanical properties. We leverage 25×25 cm samples, which is an adequate size for the manipulation of the materials.

Figure 6.15: Ranking classification accuracy on different ways of comparing materials. Comparing materials simply by measuring their distances on parameter space yields innacurate similarity rankings, as the parameter space is not orthogonal and cannot be linearly correlated with human perception of material similarity. Our Drape similarity metric provides us with a way of assessing the similarity between materials on a more perceptual-aware space, yielding more accurate rankings.



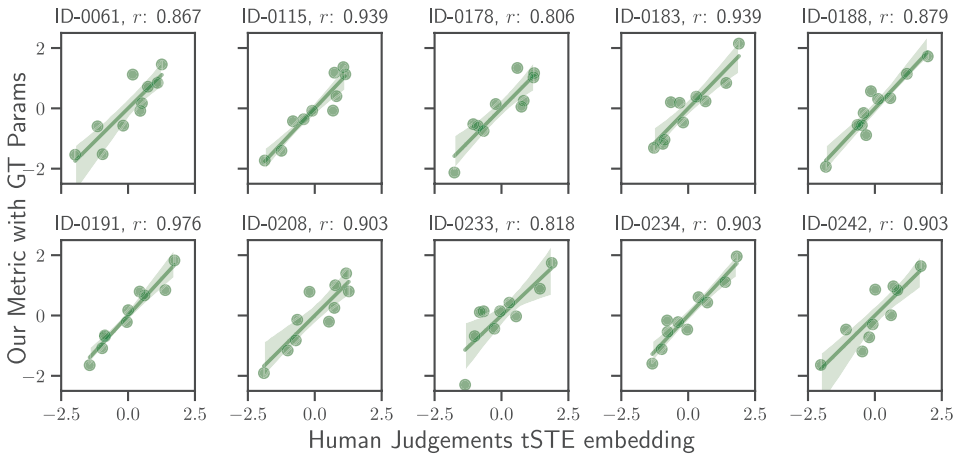
Participants. A total of 30 volunteers took part in the test. The participants had different backgrounds and diverse levels of expertise in physical manipulation of textiles. A study on their demographics can be found in Figure 6.20.

Procedure. Following previous work on human perception [Zha+18; Gar+14], we design a Two-Alternative Forced Choice (2AFC) user study in which participants are presented with triplets of physical fabric samples. We performed the experiment in a lab-controlled environment because the fabric samples had to be physically manipulated. Participants were asked to select which of two fabrics is most similar to a reference fabric. They were suggested to manipulate the samples (stretching, bending them, etc.) and to focus only on mechanical similarity, thus ignoring other factors such as optical properties (color, specularity, transparency) or irrelevant tactile feedback (softness, for instance). Material reflectance was thus ignored by participants. Volunteers were not instructed to leverage fabric motion to make their decisions but were free to dynamically manipulate the samples, which some participants did. Following recommendations in user study design for graphics [Byl+22; Dod+22], we informed the users that there are no right or wrong answers, allowed them to use their own criteria for answering the questions, and self-identify their demographics and levels of expertise. The volunteer demographics can be found on 6.20.

Each participant rated 20 triplets that were pseudo-randomly sampled following some rules: each of the 10 fabrics in our test set is shown as a reference at least twice to each participant, the order of the triplets is random, and each triplet was evaluated by at least 2 participants. Each test took between 15 to 30 minutes, depending on the participant, and was followed by an open discussion, to better understand which criteria each participant used to make their decisions. We started each test with a brief interview to register the self-reported participants' demographics [Dod+22], we continued it with the 20 perceptual comparisons and finished it with an open discussion so as to better understand which factors each participant used to make their decisions.

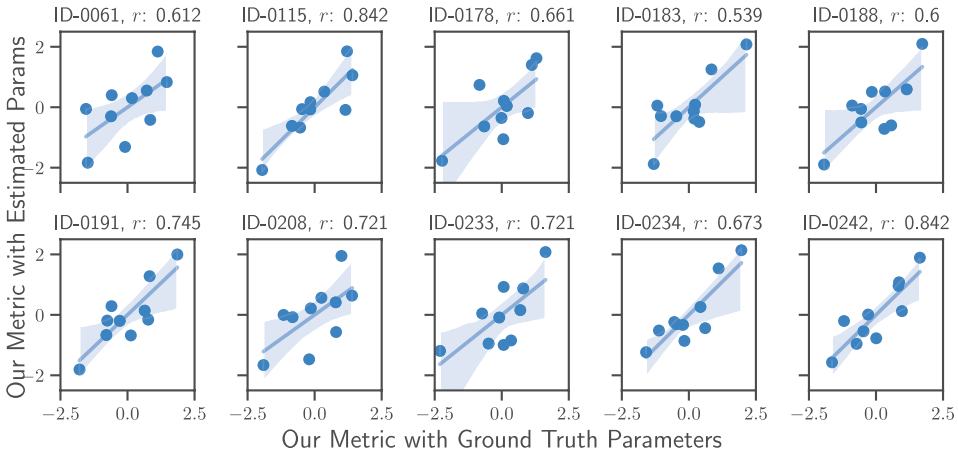
Results: Agreement 30 participants, with diverse levels of expertise with fabrics and computer simulations, volunteered for this user study. Given the same triplet, we observe an average of 86.68% agreement between our participants across all experiments and materials, suggesting that there is a perceptual understanding of fabrics mechanics that humans share. We did not observe any significant differences in agreement depending on the volunteer demographics, or level of expertise on fabric handling or simulation. This suggests that the perception of similarity of the mechanical properties of textiles may be relatively universal. When prompted, participants generally admitted that some triplets were easier to evaluate than others, either because neither candidate fabrics were similar to the reference or because both were very similar and thus difficult to make a definitive decision. Participants tended to evaluate elasticity and the shape of the fabric

Figure 6.16: Correlation between the ordering provided by the Human Judgments (x-axis) and our Drape Similarity Metric (y-axis). We plot z-scores instead of the raw distances to help visualization.



when it bends. When in doubt about those factors, a standard procedure was to assess similarity using the perceived weight of the material.

Figure 6.17: Correlation between the ordering provided by our similarity metric with the Ground Truth parameters (x-axis) and the estimated ones (y-axis). We plot z-scores instead of the raw distances to help visualization.



6.C Ablation Study: Similarity Metric Parameters

Here, we study the factors that might impact the performance of the metric: the image-based similarity metric (IM), the number of simulations necessary to account for the non-determinism of the simulation (N), and the scene (*hanging* or *stretch*).

- **Choice of** IM The image-based metric should be able to capture the subtle differences in wrinkles, folds, and overall shape of the simulated drape. While previous work has learned this metric from data [Gar+14; Lag+19], we proposed a simpler approach using existing image-based similarity metrics. Figure 6.21 (a) shows results for several deep and non-deep methods. For deep learning-based metrics, it is particularly relevant that the metric can account for local and positional differences. A *Content loss* or the *LPIPS* shows stronger performance than translation-invariant losses (e.g. *Style Loss* or *DISTS*).

- **Choice of** N We study how many simulations that are needed to account for the non-determinism of the simulation. Figure 6.21 (b) shows that $\mathrm{N} \geq 5$ is enough to obtain accurate results.

- **Choice of scene** We study which scene is best for estimating drape similarity in Figure 6.21 (c). The *stretch* scene generally provides more information than the *hanging* scene, while using both achieves the best agreement overall.

Figure 6.18: Correlation between the ordering provided by the tSTE embedding from our user study (x-axis) and our distance metric, computed using the estimated parameters (y-axis). We plot z-scores instead of the raw distances to help visualization.

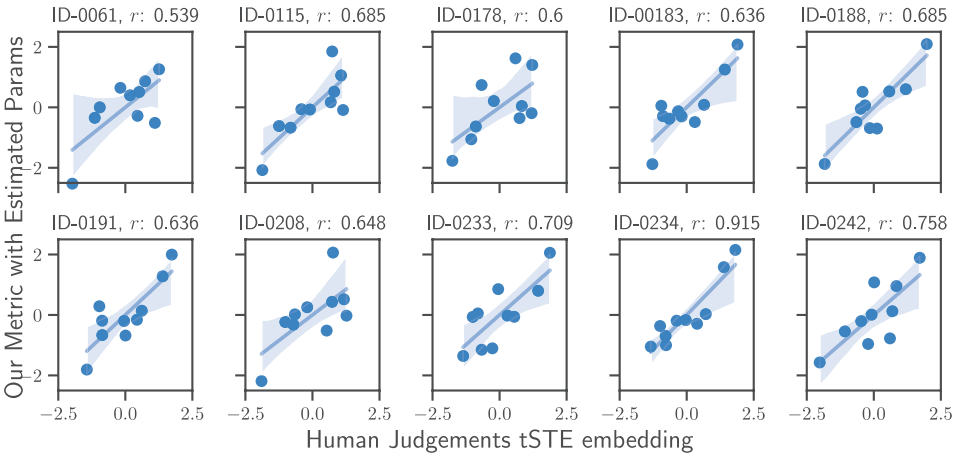


Figure 6.19: Correlation between the ordering provided by the tSTE embedding from our user study (x-axis) and the distance computed on the parameter space, for the model predictions (y-axis). We plot z-scores instead of the raw distances to help visualization.

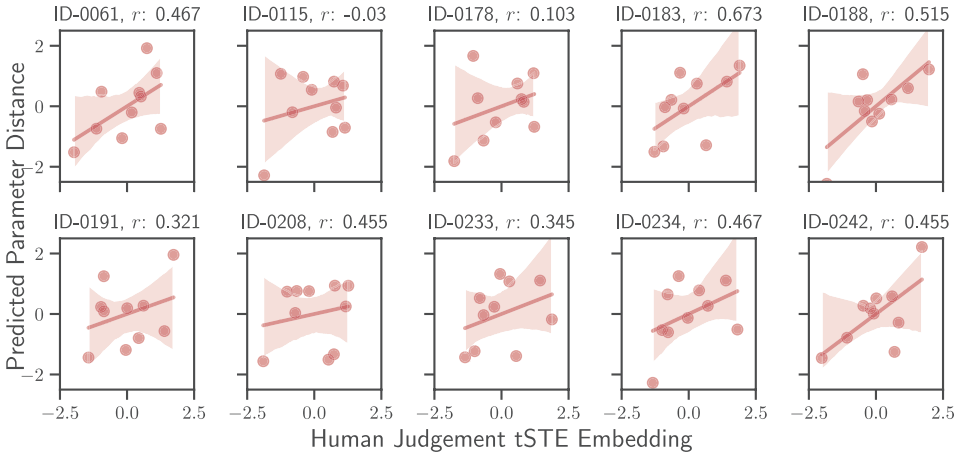


Figure 6.20: Demographics of the participants of our perceptual study. Following recommendations on user studies for computer graphics [Dod+22; Byl+22], we capture the self-identified gender, age, and education level of each participant. We also asked them to identify their level of expertise with handling physical textile materials, as well as their expertise in computer simulation and rendering of fabrics.

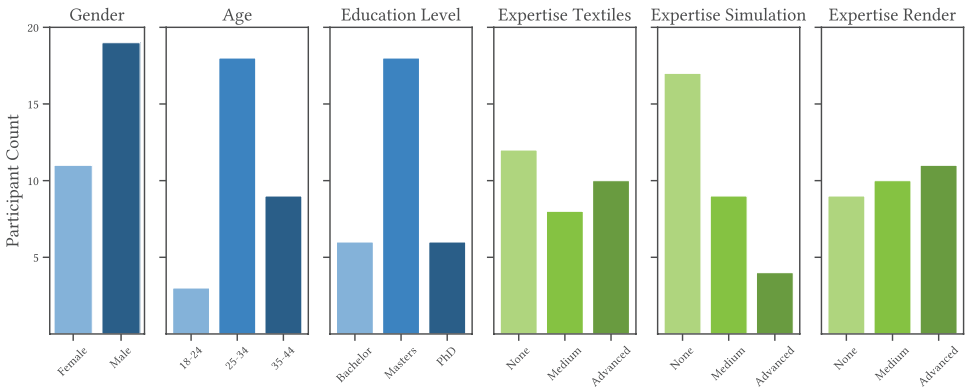


Figure 6.21: Ablation study on different factors of our similarity metric. Y-axis is Spearman correlation. (a) Results for several image-based metrics IM (we use the implementations provided in *PIQ* [KZP19]). (b) Number of simulations needed to account for the non-determinism (optimal choice is 5). (c) Impact of the scene.