Fig. 1. Garment design is complex, given the requirement to satisfy multiple specifications including target sketched fold patterns, 3D body shape, or 2D sewing patterns and/or textures. We learn a novel shared shape space spanning different input modalities that allows the designer to seamlessly work across the multiple domains. This enables to interactively design and edit garments without requiring access to expensive physical simulations at design time, and retarget the design to a range of body shapes.

Designing real and virtual garments is becoming extremely demanding with rapidly changing fashion trends and increasing need for synthesizing realistically dressed digital humans for various applications. This necessitates creating simple and effective workflows to facilitate authoring sewing patterns customized to garment and target body shapes to achieve desired looks. Traditional workflow involves a trial-and-error procedure wherein a mannequin is draped to judge the resultant folds and the sewing pattern iteratively adjusted until the desired look is achieved. This requires time and experience. Instead, we present a data-driven approach wherein the user directly indicates desired fold patterns simply by sketching while our system estimates corresponding garment and body shape parameters at interactive rates. The recovered parameters can then be further edited and the updated draped garment previewed. Technically, we achieve this via a novel shared shape space that allows the user to seamlessly specify desired characteristics across multimodal input without requiring to run garment simulation at design time. We evaluate our approach qualitatively via a user study and quantitatively against test datasets, and demonstrate how our system can generate a rich quality of on-body garments targeted for a range of body shapes while achieving desired fold characteristics. Code and data are available at our project webpage.

ACM Reference format:

1 INTRODUCTION

Developing effective tools for designing both real and virtual garments is becoming increasingly crucial. In today’s digital age, consumers are a single-click away from online clothing stores, with an increasing appetite for new fashion styles. Similarly, virtual garment design attracts increasing interest from the entertainment industry since it is a significant component of creating realistic virtual humans for movies, games, and VR/AR applications. Both of these trends are creating a demand for fast, effective, and simple tools to design, edit, and adapt garments to various body shapes.

Traditionally, designing real and virtual garments has been a complex, iterative, and time-consuming process consisting of multiple steps. First, the designer sketches the look and feel of a garment or alternatively drapes fabric on a physical dress form. Then a professional pattern maker creates the 2D garment patterns referred to as sewing patterns. A sample garment is made from the sewing patterns and tested by draping on a real or simulating on a virtual mannequin. Often the garment parameters need to be iteratively adjusted followed by redraping or resimulation until the desired look is achieved. Finally, in order to ensure an operational outfit, the mannequin is animated to see how the garment drapes across various body poses. Furthermore, the same style garment needs to be adapted for different body proportions through a process called pattern grading. This essentially requires the complex process of garment design to be repeated multiple times.
Garment design is a complex process mainly due to the fact that it operates across three different spaces, namely 2D sketches for initial design, 2D sewing patterns and pattern selection for parameterized modeling, and 3D garment shape to model the interaction between garments and subject bodies producing on-body garments (see Figure 2). Much of pattern making involves using various rule-based specialized cuts and stitches (e.g., darts, pleats, yokes) to achieve desired folds on the final draped garment. Note that a particularly challenging scenario is designing free-flowing garments where the characteristic patterns arise due to the interaction between fabric hidden behind the creases and boundary conditions induced by the underlying body shape. To aid with such challenging scenarios, an ideal computational tool should allow the designer to freely navigate across the three design spaces and effectively capture the interaction between them.

Designing such a unified treatment of the three spaces has remained elusive due to several challenges. First, interpreting 2D garment sketches requires a good understanding of what shapes and folds are possible in 3D. Although the problem appears to be badly ill-conditioned, as humans, we regularly use our experience of typical garment folds and looks to ‘regularize’ the problem and interpret artist sketches. Second, the relation between the 2D garment parameters and the final 3D shape of the garment is highly non-linear depending not only on the shape of the garment itself but also its material properties, the pose, and the shape of the 3D body it is being draped on. This necessitates a computationally heavy cloth simulation process to visualize patterns arising out of garment folds, creases, and ruffles. Finally, modeling in 3D needs to ensure the physical plausibility of the garment by mapping the 3D designs to (near-) developable 2D sewing patterns.

We present a data-driven approach that overcomes the above challenges by learning a shared shape space that, for the first time, unifies 2D sketches; parameters of 2D sewing patterns, garment materials, and 3D body shapes; and the final draped 3D garments. We achieve this by jointly training multiple encoder-decoder networks that each specializes at linking pairs of representations (e.g., recovering garment and body parameters from a sketch or recovering the draped garment shapes from the parameters) while operating at a common embedding. To train these networks, we create a large scale synthetic dataset. Specifically, we first define a set of parameterized garment types (shirt, skirt, and kimono) and generate different garments by sampling this representation. Then, we simulate each garment on a set of 3D body shapes sampled from a deformable body model. Finally, for each of these simulated examples, we generate 2D sketches using non-photorealistic rendering. Thus, each example triplet in our dataset includes (i) a 2D sketch, (ii) garment and body parameters, and (iii) the resultant draped 3D shape of the garment. Subsequently, by jointly training multiple encoder-decoder networks via a novel multimodal loss function, we learn a common embedding that can be queried using any of the different modalities. Note that our network assumes that the 2D pattern topology remains fixed and does not allow transitioning across different pattern families. This remains a current limitation and an open challenge to explore through future research.

The learned shared shape space enables several applications by linking the different design spaces. For example, starting from an input 2D sketch, we can (i) automatically infer garment and body shape parameters; (ii) predict the resultant 3D garment shapes from these parameters without going through an expensive cloth simulation process; (iii) directly texture the final garments using the linked 2D sewing pattern parameterization; (iv) sample from or interpolate in the latent space to generate plausible garment variations; or (v) retarget 2D garment patterns to new body shapes such that the resultant draped garments retain the original fold characteristics. At any stage of the design and retargeting process, the garment and body parameters inferred by our method can be provided to a cloth simulator to generate the physically accurate shape of the garment on the 3D body. Figure 1 shows several examples.

We qualitatively and quantitatively evaluate our approach against groundtruth test data, and demonstrate our interactive garment sketching for various applications. We also provide comparisons with alternative methods and show favorable performance (see Section 5). In summary, our main contributions are: (i) a method that learns a joint embedding of different design spaces for a given garment type; (ii) inferring garment and body parameters from single sketches; (iii) estimating the 3D draped configurations of the garments from the garment and body parameters to enable an interactive editing workflow; and (iv) facilitating fold-aware pattern grading across different body shapes via a novel garment retargeting optimization.

2 RELATED WORK

Garment modeling. Traditional garment design is a complex process. Starting with initial 2D sketches of the desired garment, the designer creates the necessary flat sewing patterns which are then stitched together into a garment. Physical draping or physics-based simulation is used to infer the final shape of the garment. While there exist many professional tools (e.g., Optitex [opt 2018], Marvelous Designer [mar 2018]) to assist designers with the process, there has been a significant effort in the research community to propose alternative computational tools.

Many of the previous works have specifically focused on bridging the gap across two different design spaces. For example, Berthouzoz et al. [2013] focus on automatically parsing 2D garment patterns to
Interactive Garments

Fig. 3. Given an input sketch, our network infers both the 2D garment sewing patterns (in blue) and the draped 3D garment mesh together with the underlying body shape. Edits in the 2D sewing patterns (e.g., shorter sleeves, longer shirt as shown in red) or the sketch (bottom row) are interactively mapped to updated 2D and 3D parameters. The 3D draped garment inferred by our network naturally comes with uv coordinates, and thus can be directly textured. The network predictions can be passed to a cloth simulator to generate the final garment geometry with fine details. Finally, the designed garment can be easily retargeted to different body shapes while preserving the original style (i.e., fold patterns, silhouette) of the design.

3D, while other methods have focused on modeling 3D garments via input sketches. Several sketch-based interfaces [Decaudin et al. 2006; Robson et al. 2011; Turquin et al. 2004] have been proposed where silhouette and fold edges in an input sketch are analyzed to create a 3D garment. However, they assume the input sketch is provided with respect to a given 3D mannequin and use the body shape of the mannequin to lift the sketch to 3D. The recent approach of Jung et al. [2015] provides a more general tool for modeling 3D developable surfaces with designed folds but require multi-view sketch input (e.g., frontal, side, and optionally top). The freeform surface modeling tool, BendSketch [Li et al. 2017], is capable of modeling plausible 3D garments from user sketches but has no notion of the corresponding 2D sewing patterns. Recently, Li et al. [2018] present a modeling system where the user can sketch different types of strokes on an existing 3D draped garment to denote different types of desired folds. The system then extends and optimizes the 2D garment patterns such that the draped garment exhibits the desired folds. While this system provides impressive results, it assumes an initial 3D draped garment to be provided. In contrast, our system automatically infers the parameters of the garment and the body from an input sketch and maps them to the final 3D draped shape of the garment. We achieve this goal by proposing a deep learning based solution which draws inspiration from recent methods that use deep learning to infer 3D shapes from 2D sketches [Han et al. 2017] or to interactively control shape variability [Umetani 2017]. Once the garment parameters are predicted by our method from a single sketch, the method of Li et al. [2018] or the rule-free method of Wang et al. [2018] can be used to refine cuts on the resulting 2D garment patterns.

In another line of work, researchers have proposed to use other types of input such as images or RGBD scans to model garments. Given an input image, Zhou et al. [2013] first estimate the 3D body shape in a semi-automatic fashion and then lift extracted silhouette edges of the garments to 3D. Jeong et al. [2015] use an image of a garment on top of a mannequin as input and detect silhouette edges and landmark points to create a 3D garment. Chen et al. [2015] model garments from RGBD data as a combination of 3D template components based on a set of rules. Given a database of garment templates, Yang et al. [2016] propose a relatively complex pipeline to determine the parameters of these garments to match an input image. More recently, Daněřek et al. [2017] present a deep learning based method which predicts the deformation of a garment from a reference 3D garment given an input image. While the proposed approaches provide plausible garments, they often fail to capture the details, i.e., the exact fold patterns observed in the input. In contrast, our goal is to be able to reconstruct such folds to enable realistic design and editing of garments.

Garment editing and retargeting. In addition to modeling garments from scratch, several methods have been proposed for editing the shape and appearance of them. Umetani et al. [2011] propose an interactive editing system that enables bi-directional editing between 2D garment patterns and 3D draped forms. Meng et al. [2012] explore apparel resizing with flexible shape control. Kwok et al. [2016] investigate stylizing tight-fitting garments. Bartle et al. [2016] present a method to map 3D edits to a garment to plausible 2D garment patterns. In contrast, we support a multi-modal design and editing paradigm, specifically focusing on modeling the desired folds and silhouettes of a garment.

Retargeting the style of an existing garment to body shapes with different proportions is a specific form of editing that has attracted...
special attention. Brouet et al. [2012] formulate a constrained optimization framework to transfer garments across different 3D characters. Other approaches use data-driven methods [Guan et al. 2012; Xu et al. 2014] to replace the expensive physical cloth simulation process and present retargeting examples. In our work, we perform garment retargeting via a novel optimization procedure that directly operates at the joint embedding of different garment design spaces and can be used to transfer across 100s of shape variations while ensuring that desired fold characteristics are preserved.

**Garment capture.** While garment modeling approaches aim to generate realistic garments, capture methods focus on faithfully reconstructing the garment observed in the input data. They accomplish this task often by utilizing more complex capture setups. Pritchard et al. [2003] present one of the earlier approaches where the geometry of a garment is reconstructed from a stereo image pair. Several follow-up works have instead used a multi-view capture setup to reconstruct garments with color-coded patterns [Scholz et al. 2005; White et al. 2007]. Bradley et al. [2008] have eliminated the need for a color-coded pattern and presented a multi-view markerless motion capture system for garments. Their follow-up work [Popa et al. 2009] aims to add details due to wrinkles and folds to the coarse meshes captured from the multi-view video input. Wang et al. [2010] solve a similar problem of augmenting coarse 3D garment geometry with wrinkles using a data-driven approach. In a separate line of work, given a 2D pattern and 3D contour curve, Rohmer et al. [2011] interpolates the curve in a procedural manner to generate folded paper geometry. The more recent approach of Neophytou et al. [2014] analyzes a set of 3D scans to learn a deformation model for human bodies and garments where garments are represented as the residual with respect to the body shape. Finally, the ClothCap [Pons-Moll et al. 2017] system takes a 3D scan sequence as input and captures both the body and the garment shape assuming weak priors about where a specific type of garment is expected to be with respect to the body. All these methods perform a faithful reconstruction of the garments including the fold and wrinkle geometry but rely on multi-view input or alternate 3D information. In contrast, our method performs a similar prediction using a single sketched image as input and allows for subsequent multi-modal refining in the garment domain and/or the mannequin body shape.

3 APPROACH

3.1 Overview

Traditional garment design workflows involve interacting with one or more design domains, namely: (a) the 2D design sketch \( S \) that can be used to indicate a desired look and feel of a garment by specifying silhouette and fold lines; (b) the parameter domain \( P \) that allows the designer to specify both pattern parameters (i.e., size/material of sewing patterns) and body parameters (i.e., shape of the human body); and (c) the 3D draped configuration \( M \) (i.e., the 3D mesh) that captures the final garment shape on the target human body with a garment sized according to its 2D pattern parameters.

The above-mentioned domains have complementary advantages. For example, sketches provide a natural option to indicate visual characteristics of the folds such as density of folds, silhouette, etc.; parameters are effective to indicate direct changes to garment edits and/or specify target body shapes; while, the draped shape helps to generate previews under varying texture patterns, camera and/or illumination settings. By providing the designer with the ability to indicate target specifications via multiple modalities, we want to exploit the complementary advantages offered by the different domains to enrich the design process.

The above spaces, however, have very different dimensions making it challenging to robustly transition from one domain to another, or accept inputs across multiple modalities. In other words, a traditional data-driven method can easily overfit to the example datasets and result in unrealistic results in case of new test data. For example, a learned network to help transition from \( S \to M \) easily leads to over-fitting as seen on test sketch input in Figure 4 (see Section 5). More importantly, such an approach does not give the designer access to the pattern and/or body parameters to edit.

![Fig. 4. While a network trained directly to infer the draped garment from an input sketch overfits to training data, learning a joint (latent) shape space across different modalities leads to better generalization.](image)

Instead, we propose to learn a shared shape, i.e., latent, space by jointly learning across the three domains using a novel cross-modal loss function (Section 3.3). Our key observation is that the shared latent space regularizes the learning problem by linking the different domains. From a usage point of view, the artist enters the design space via a sketch, and then continue making further changes by directly editing the inferred pattern and/or body parameters. The designer can now create garments via a multimodal interface by seamlessly indicating sketch behavior, garment or body parameters, or retexturing (see Figure 3). A garment, thus designed, can then be easily remapped to a range of other body shapes, facilitating pattern grading. To ensure the original sketched folding behaviors do not get lost in the new draped garments adapted for the different body shapes, we present a novel retargeting method that formulates an optimization in the shared latent space (Section 3.4). Before describing the methods in detail, we next introduce the specific representation choices used in this work.

3.2 Parameter Space

In this work, we tested on three different garment types namely shirts, skirts, and kimonos. We parameterize each garment type using a small number of key dimensions (e.g., length and width of sleeve for shirt, length of the waistline for skirt) and material properties, i.e., stretch, bend, and shear stiffness (Figure 5). Specifically, the number of parameters for each garment types were: 4 for kimono, 9 for shirt, and 11 for skirt, respectively. We collect all the garment parameters (including dimension parameters and
In order to generate a dataset of training examples for a given garment type, we first randomly sample garment parameter instances from $G$ to generate the 2D patterns. With a target pose, we then sample one of the $2k$ female body shapes in the FashionPose dataset [Lassner et al. 2017] to generate samplings of $P$ resulting in a total of 8000 combinations. We then simulate combinations of garment samples over body samples using the cloth simulator in FleX [nvi 2018] (see Figure 8). This results in draped garment meshes, which we refer to as the mesh $M$. Given a collection of such meshes ($M$) corresponding to the same type of garment, we obtain a compressed representation by performing Principal Component Analysis (PCA) and represent each garment mesh using the first $K$ eigenvectors.

In FleX [nvi 2018] (see Figure 8), this results in draped garment meshes, which we refer to as the mesh $M$. Given a collection of such meshes ($M$) corresponding to the same type of garment, we obtain a compressed representation by performing Principal Component Analysis (PCA) and represent each garment mesh using the first $K$ eigenvectors.

3.3 Shared Shape Space

Given the 3 different modalities ($P, M, S$) for each example, our goal is to learn a joint shape space shared between the 3 modalities. Given a sketch or a parameter sample, and (iii) the reconstruction error of a parameter sample from itself in an auto-encoder fashion. Thus, the combined loss function is defined as:

$$L(P, M, S) = \omega_1 ||P - f_{L2P}(f_{S2L}(S))||_2 + \omega_2 ||M - f_{L2M}(f_{S2L}(S))||_2 + \omega_3 ||M - f_{L2M}(f_{P2L}(P))||_2 + \omega_4 ||P - f_{L2P}(f_{P2L}(P))||_2,$$

where $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ denote the relative weighting of the individual losses. We set these weights such that the average gradient of each loss is at the same scale (in our experiments $\omega_1 = \omega_2 = 40 \omega_3 = 40 \omega_4$). Empirically the consistency terms (the last three terms) make a significant difference on the quality of the network prediction on test data.

3.4 Garment Retargeting

One of the most common tasks in real or virtual garment design is retargeting, i.e., adapting a particular garment style to various body shapes. Given a garment $G$ designed for a particular body style $B$, the goal of the retargeting process is to identify a new set of garment parameters $G'$ for a new body shape $B'$ such that the look and feel of the draped garments on both body shapes are similar. Naively using the same set of garment parameters on a new body shape does not preserve the original style as shown in Figure 17. On the other hand, deforming the draped garment in 3D to match the desired style does not ensure a mapping to a valid configuration of sewing patterns.

![Fig. 6. We learn a shared latent shape space between (i) 2D sketches, (ii) garment and body shape parameters, and (iii) draped garment shapes by jointly training multiple encoder-decoder networks.](image)
within the silhouette of sewing patterns using centroidal Voronoi body shapes. Note that we discarded any pairs of sampled clothes triangulation. We then simulate each sampled garment on varying set of garment parameters and generate an isotropic triangle mesh for a given garment type, i.e., shirt, skirt, or kimono. When generating our training dataset, for a specific garment type, we first sample a set of garment parameters and generate an isotropic triangle mesh

\[ \mathcal{E}(G' | G, B, B') = \| \mathcal{L}(G, B, B') - \mathcal{L}(G', B, B') \| \]

(2)

Since both \( \mathcal{G} \) and the mappings \( \mathcal{L}_{2M}, \mathcal{L}_{2L} \) are learned via differentiable networks, we can efficiently compute the gradient of the objective function during the optimization. In our experiments, we used an L-BFGS solver to optimize the above energy.

4 IMPLEMENTATION DETAILS

In the following, we provide details about the data generation process and the network architecture.

**Data generation.** When generating our training dataset, for a given garment type, i.e., shirt, skirt, or kimono, we first sample a set of garment parameters and generate an isotropic triangle mesh within the silhouette of sewing patterns using centroidal Voronoi triangulation. We then simulate each sampled garment on varying body shapes. Note that we discarded any pairs of sampled clothes and bodies whenever the draped garment was too tight or too loose after simulation. The cloth simulator we use, i.e., FleX, is particle-based and is sensitive to the edge lengths of the mesh being simulated. Thus, we fix the average edge length across all samples of the garment, leading to meshes with varying topology and face count. In order to ensure a consistent mesh topology to perform PCA on draped garments, we use one of the simulated examples as a reference mesh topology and remesh the remaining examples. Specifically, we locate the vertices of the reference template in its associated triangle in a common sewing pattern space and compute its position in every other simulated example via barycentric coordinates. In order to balance efficiency and accuracy of simulations, we used a mesh template with about 27.5k vertices for the kimono dataset, 12.3k vertices for the skirt dataset, and 5k for the skirt dataset. Empirically, we found these resolutions to consistently produce fine detail wrinkles during simulation. The simulation of 8000 samples for each garment type took about 60 hours to generate with 2 Nvidia Titan X GPUs.

**Network architecture.** The encoder and decoder networks we train to learn a shared shape space, i.e., latent space, are composed of linear blocks which are linear layers followed by Rectifying Linear Units (ReLU) and are followed by a projection layer to project the input to a low-dimensional space. The encoder is a Siamese network with a simple architecture provided in the TorchVision library [tor 2018] and uses the DenseNet-161 paper. We perform data augmentation in the resulting sketches by blurring, etc. All sketches are centered and cropped into a 224 \( \times \) 224 square patch. We extract a 2208-dimensional feature vector for each patch via DenseNet [Huang et al. 2017] (we use the DenseNet-161 architecture provided in the TorchVision library [tor 2018] and use the output of the fourth dense block as our feature vector). Note that while we used DenseNet to provide feature descriptor for the input sketch, other descriptors such as VGG [Simonyan and Zisserman 2014] or ResNet [He et al. 2016] can alternatively be used in our setup.

Fig. 7. We train a Siamese network \( \mathcal{G} \) that learns an embedding of draped garment shapes \( (M) \). The distance between a pair of draped garments \( (M, M') \) in this learned embedding space is similar to the distance between the HOG features of the corresponding sketches. Once trained, the loss function can be differentiated and used for retargeting optimization.

Fig. 8. To generate synthetic data, we first shrink the body of the mannequin in rest pose into its skeleton and let the triangle mesh of the 2D sewing patterns drape naturally over it. After the draping converges, we stitch the boundary and inflate the skeleton back to its original shape. We further animate the mannequin into the target body pose to generate our final simulation result.

![Image](image-url)
Table 1. Starting with an input sketch (top) or a set of garment and body shape parameters (bottom), we report the average L2 error in the estimated draped garment (both in terms of PCA basis and vertex positions), body shape (both in terms of SMPL parameters and vertex positions), and the garment parameters. All the parameters are normalized to the range [0, 1].

<table>
<thead>
<tr>
<th>From sketch</th>
<th>Garment mesh</th>
<th>Body shape</th>
<th>Garment parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>Testing set</td>
<td>Training set</td>
<td>Testing set</td>
</tr>
<tr>
<td>Shirt</td>
<td>1.99%</td>
<td>1.58%</td>
<td>4.21%</td>
</tr>
<tr>
<td>Skirt</td>
<td>1.76%</td>
<td>1.14%</td>
<td>2.38%</td>
</tr>
<tr>
<td>Kimono</td>
<td>2.28%</td>
<td>1.70%</td>
<td>5.45%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From parameter</th>
<th>Garment mesh</th>
<th>Body shape</th>
<th>Garment parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set</td>
<td>Testing set</td>
<td>Training set</td>
<td>Testing set</td>
</tr>
<tr>
<td>Shirt</td>
<td>1.56%</td>
<td>1.16%</td>
<td>3.57%</td>
</tr>
<tr>
<td>Skirt</td>
<td>1.29%</td>
<td>0.98%</td>
<td>2.03%</td>
</tr>
<tr>
<td>Kimono</td>
<td>1.80%</td>
<td>1.58%</td>
<td>5.21%</td>
</tr>
</tbody>
</table>

Unit (RELU) activations and batch normalization. Specifically, the encoder, \( F_{52L} \), takes as input a 2208-dimensional feature vector of a sketch and maps it to the \( K = 100 \) dimensional latent space with 10 linear blocks (the output dimension size is kept fixed in the first 6 blocks and gradually decreased to 1000, 500, 200, and 100 in the remaining 4 blocks). The encoder, \( F_{P2L} \), takes as input a \( p \)-dimensional parameter vector representing the garment and the body shape (\( p = 22 \) for \( \text{shirt} \), \( p = 17 \) for \( \text{skirt} \), \( p = 24 \) for \( \text{kimono} \)) and maps it to the latent space with 6 linear blocks (the output dimension size is kept fixed in the first block and increased to 100 in the second block). The decoder, \( F_{L2M} \), takes as input the \( K = 100 \) dimensional latent space vector and maps it to the 200-dimensional PCA basis that represent the draped garment shape. This decoder consists of 6 linear blocks (the output size of the first two blocks is 100 and the output size of the remaining blocks are 200). Finally, the decoder, \( F_{L2P} \), takes as input the \( K = 100 \) dimensional latent space vector and maps it to the parameters of the garment and the body shape. This decoder consists of 6 linear blocks, where the first 5 blocks keep the output dimension size fixed and the last block changes the output size based on the garment type.

We jointly train the encoder-decoder architectures for 20k epochs with a learning rate of 0.1 and batch size of 64. We use stochastic gradient descent for network backpropagation. We used PyTorch for implementation and the proposed network takes about 3 hours to train on each dataset with one Nvidia Titan X GPU.

**Retargeting.** For retargeting garments across different body shapes, we train a Siamese network that learns an embedding of the draped garments such that the distance between two draped garments is similar to the distance between their corresponding sketches. This Siamese network takes as input a 200-dimensional PCA vector and maps it to an 100-dimensional embedding space with 6 linear blocks (the output dimension size is kept fixed in the first 5 blocks and decreased to 100 for the last block). We train this network for 20k epochs with a learning rate of 0.03 and batch size of 64. We use stochastic gradient descent for network backpropagation.

We refer the reader to the supplementary material for a detailed network architecture configuration.

5 **EVALUATION**

We evaluate our method qualitatively on real and synthetic data and quantitatively on synthetic data. We split our synthetic dataset into training (95%) and testing sets (5%) such that no garment and body parameters are shared across the splits. For evaluations on the synthetic data, we use 200 samples from the test set for each garment type. We note that given an input sketch, our method estimates the corresponding garment and body shape parameters as well as the shape of the draped garment. In a typical design workflow, once satisfied with the design, we expect the designer to perform a cloth simulation using the estimated garment and body shape parameters to obtain the final and accurate look of the garment. The draped garment shape predicted by our network closely resembles the simulation result performed by using the predictions of the garment and body parameters. Thus, we only provide the final simulated results. Note that since we used a particle-based simulator, the simulation output cannot be ‘touching’ each other in our setup, this safety margin was 4mm with respect to an average body height.
Fig. 10. For each sketch, we show our output with/without texture; NPR rendering of our output and the nearest sketch retrieved from our database. As highlighted in orange, our result plausibly captures the folds provided in the input sketch.
Interactive Garments

Fig. 11. For each reference image, we ask users to draw the corresponding sketch that we provide as input to our method. Our method generates draped garments that closely resemble the reference images.

of 1.8m. In practice, we observed some minor interpenetrations which were removed via resimulation with estimated parameters as shown in Figure 9. In Figure 10, for each input synthetic sketch, we show the simulated results with the garment and body shape parameters predicted by our method. Since our network can predict the corresponding 2D sewing pattern parameters as well, we can easily generate uv coordinates for the draped garment and texture it. To show that our network does not learn to memorize the training examples, we also show the nearest neighbors retrieved from our training dataset using the DenseNet features of sketches. As highlighted in the figure, while the NPR renderings of our results closely resemble the input sketch, the nearest neighbors fail to capture many folds present in the input sketch.

In Figure 11, we provide examples on real images to test the generality of our approach. For 6 different reference images, we ask different users to draw the corresponding sketch that is provided as input to our method. As shown in the figure, our method is able to generate the draped garments that closely follow the reference image and the input sketches.

We demonstrate the consistency of the uv parameters generated by our method by texturing the same garment type with varying parameters draped on varying body shapes (see Figure 12). Our method generates uv parameters that are free of distortions compared to generic solutions.

Quantitative evaluations. We quantitatively evaluate the performance of the different mappings learned by our network. Specifically, starting either with an input sketch (or a set of garment and body shape parameters), we first map the input to the latent space via \( f_{S2L} \) (or \( f_{P2L} \)). Then, we measure the error in the predicted draped garment mesh (the output of \( f_{L2M} \)) and the estimated garment and body shape parameters (the output of \( f_{L2P} \)). For the draped garment, we report the average L2 error both on the PCA coefficients and the vertex positions, similarly, for the body shape, we report the average L2 error both on the SMPL shape parameters and the vertex positions. We normalize all the parameters and the garment and
Our method generates consistent uv parameters across different instances of the same garment type. In this example, the alignment between the texture patterns can be best seen in the neck region.

Fig. 13. Given two kimono designs shown on the left and the right, we interpolate between their corresponding descriptors in the shared latent space. For each interpolation sample, we show the 3D draped garments inferred by our network and the NPR rendering of the simulated mesh from the corresponding sewing parameters. Our shared latent space is continuously resulting in smooth in-between results.

Joint shape space evaluation. In order to demonstrate the benefit of jointly training a shared latent space across three modalities, we train an alternative single encoder-decoder network composed of the mappings \(f_{S2L}\) and \(f_{L2M}\). As shown in Figure 4, this direct mapping overfits to the training data and is not able to generalize. In contrast, jointly training for additional mappings regularizes the problem and leads to better performance during test time.

Our learned shared latent space is compact and smooth, as shown in Figure 13. When we linearly interpolate between two samples in the latent space, we obtain intermediate results that change smoothly both in terms of 3D predict mesh and garment sewing parameters.

Comparison. In Figure 15, we compare our approach with one example from Figure 8 of Yang et al. [2016]. Given the reference image in this example, we ask a user to provide the corresponding sketch that is provided as input to our method. The estimate draped garment shape by our method is of similar quality but is generated at interactive rates once the network is trained by our skirt dataset in contrast to the instance-wise computation-heavy approach of Yang et al. [2016].

Retargeting evaluation. As shown in Figure 17, draping the same garment on different body shapes do not preserve the style of the garment. Our retargeting optimization, in contrast, identifies a new set of garment parameters that would preserve the style of the original garment on a body shape.

Our retargeting optimization uses a distance measure between draped garments based on the Siamese network introduced in Section 3.4. We show a tSNE visualization [van der Maaten and Hinton 2008] of the embedding learned by this network in Figure 19. For each embedding, we provide the sketches corresponding to various random samples showing that similar sketches get clustered.
Interactive Garments

Fig. 16. Given a reference garment style on the left, we retarget it to various body shapes. For each example, we also provide the NPR renderings of the draped garments.

Fig. 17. Given an initial garment design, draping the same garment on a different body shape does not preserve the style of the design. In contrast, our retargeting method optimizes for a new set of garment parameters that result in the same look of the garment on a new body shape.

User study. Finally, we conduct a user study to evaluate how closely our results resemble the input sketches. Specifically, given two input sketches \((S_i, S_j)\), we show the corresponding draped garment shapes \((M_i, M_j)\) predicted by our network and ask the user to pair the sketches to the garments (e.g., \(S_i\) should be paired with \(M_i\)). If the network output successfully captures the folds in the input sketch, we expect the accuracy of such pairings to be high. However, if \(S_i\) and \(S_j\) are similar to begin with, the pairing becomes ambiguous. We generate 400 pairing queries for each garment type and ask 13 Amazon Mechanical Turk users to answer each query. We plot the accuracy of the pairings vs. the similarity of the given sketches. The users are accurate with pairings unless the input sketches are very similar.

Interactive user interface. Based on the shared shape space we learn across different design modalities, we implement an interactive user interface (see Figure 20). In this interface, the user can interactively edit the input sketch, the garment or the body shape parameters, or the uv coordinates and interactively visualize the corresponding draped garment. We refer to the supplementary video which shows each of these editing options.

6 CONCLUSION
We presented a data-driven learning framework for obtaining a joint shape space linking 2D sketches, garment (2D sewing patterns and material properties) and human body specifications, and 3D draped garment shapes in the context of garment design. The learned shape space enables a novel multimodal design paradigm that allows users to iteratively create complex draped garments without requiring expensive physical simulations at design time. We also utilize the
learned space to formulate an optimization that allows designed garments to be retargeted to a range of different human body shapes while preserving the original design intent. Finally, we evaluated our method, both quantitatively and qualitatively, in different usage scenarios and showed compelling results.

**Limitations and future work.**

Our method has certain limitations that open up interesting future directions. For each garment type, we have a pre-defined set of 2D sewing parameters. Thus, we cannot represent garment shapes that are not covered by this set as in the inset example. Additionally, the choice of garment simulator also limits the pattern complexity and material variance that can be generated in the dataset. As a result, the network learns to produce pattern and material variations only as observed in the data. As simulation frameworks get more powerful, we expect that the practicability of our method for fashion design and modeling will improve.

Currently, our method does not handle pose variation, we assume the garments are designed for a body at a fixed target pose. In the future, we plan to expand the shape space by also regressing across common body pose variations. In Figure 21, we show preliminary results in this direction. We augment our dataset by considering an additional degree of freedom for body pose that interpolates between different arm configurations. Then, given the input sketches in the top row, we infer the draped garments and body shape and pose parameters shown in the bottom row. These results indicate
that by augmenting the dataset, learning pose variations is possible. The challenge, however, is how to effectively sample the pose configurations space.

In this work, we learned shape spaces specialized to different garment types (shirt, skirt, and kimono). However, the trained network are garment-specific and cannot be reusable for new garment types. A difficult research question is how to unify these different garment-specific shape spaces via a common space. This is challenging given the complex discrete and continuous changes necessary to transition from one garment type to another. One possibility is to perform interactive exploration where the user annotates additional cut/fold lines as recently demonstrated in [Li et al. 2018]. Note that the lack of discrete parameters also limited the design complexity of our system and hence we cannot support operations such as insertion of necklines, collars, or cuffs. Finally, we would like to explore the effectiveness of the joint learning in other design contexts involving multimodal inputs.

Acknowledgement. We thank the reviewers for their comments and suggestions for improving the paper. The authors would also like to thank Dr. Paul Guerrero for helping with user study and Dr. Michal Lukáč for the video voice-over. This work is in part supported by a Microsoft PhD fellowship program, a Rabin Ezra Trust Scholarship, a Google faculty award, a ERC Starting Grant SmartGeometry (StG-2013-335373), and UK Engineering and Physical Sciences Research Council (grant EP/K023578/1).

REFERENCES


