# TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style

Chaitanya Patel* Zhouyingcheng Liao* Gerard Pons-Moll
Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
{cpatel, zliao, gpons}@mpi-inf.mpg.de

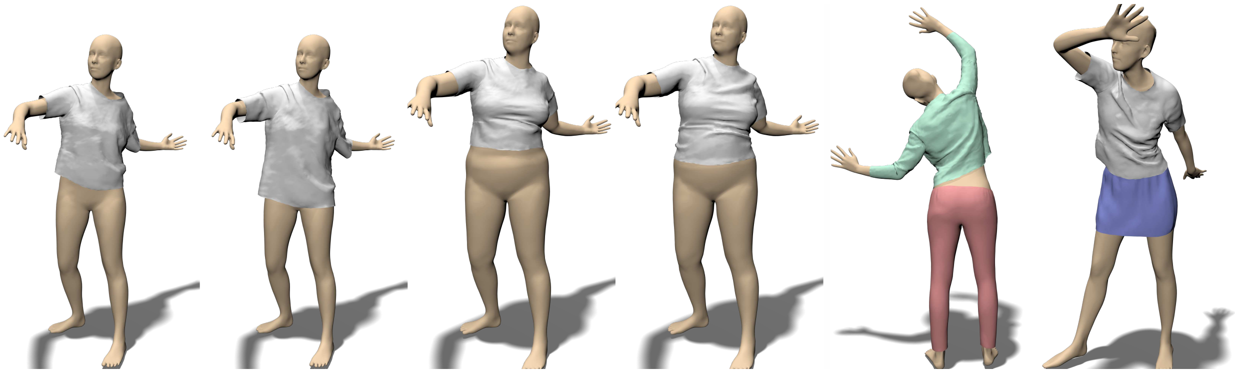

Figure 1: We present TailorNet, a model to estimate the clothing deformations with fine details from input body shape, body pose and garment style. From the left: the first two avatars show two different styles on the same shape, the following two show the same two styles on another shape, and the last two avatars illustrate that our method works for different garments.

## Abstract

*In this paper, we present TailorNet, a neural model which predicts clothing deformation in 3D as a function of three factors: pose, shape and style (garment geometry), while retaining wrinkle detail. This goes beyond prior models, which are either specific to one style and shape, or generalize to different shapes producing smooth results, despite being style specific. Our hypothesis is that (even non-linear) combinations of examples smooth out high frequency components such as fine-wrinkles, which makes learning the three factors jointly hard. At the heart of our technique is a decomposition of deformation into a high frequency and a low frequency component. While the low-frequency component is predicted from pose, shape and style parameters with an MLP, the high-frequency component is predicted with a mixture of shape-style specific pose models. The weights of the mixture are computed with a narrow bandwidth kernel to guarantee that only predictions with similar high-frequency patterns are combined. The style variation is obtained by computing, in a canonical pose, a subspace of deformation, which satisfies physical constraints such as inter-penetration, and draping on the body. TailorNet delivers 3D garments which retain the wrinkles from the physics based simulations (PBS) it is learned from, while running more than 1000 times faster. In contrast to classical PBS, TailorNet is easy to use and fully differentiable, which is crucial for computer vision and learning algorithms. Several experiments demonstrate TailorNet produces more realistic results than prior work, and even generates temporally coherent deformations on sequences of the AMASS [34] dataset, despite being trained on static poses from a different dataset. To stimulate further research in this direction, we will make a dataset consisting of 55800 frames, as well as our model publicly available at https://virtualhumans.mpi-inf.mpg.de/tailornet/.*

## 1. Introduction

Animating digital humans in clothing has numerous applications in 3D content production, games, entertainment and virtual try on. The predominant approach is still physics based simulation (PBS). However, the typical PBS pipeline requires editing the garment shape in 2D with patterns, manually placing it on the digital character and fine tuning parameters to achieve desired results, which is laborious, time consuming and requires expert knowledge. Moreover, high quality PBS methods are computationally expensive, complex to implement and to control, and are not trivially

*Equal contribution.

differentiable. For computer vision tasks, generative models of clothed humans need to be differentiable, easy to deploy, and should be easy to integrate within CNNs, and to fit them to image and video data.

In order to make animation easy, several works learn efficient approximate models from PBS complied off-line. At least *three factors* influence clothing deformation: body pose, shape and garment style (by style we mean the garment geometry). Existing methods either model deformation due to pose [10, 29] for a fixed shape, shape and pose [46, 18] for a fixed style, or style [58] for a fixed pose. The aforementioned methods inspire our work; however, they do not model the effects of pose, shape and style jointly, even though they are intertwined. Different garments deform differently as a function of pose and shape, and garment specific models [46, 29, 17, 10] (1 style) have limited use. Furthermore, existing joint models of pose and shape [17, 8, 18] often produce over-smooth results (even for a fixed style), and they are not publicly available. Consequently, none of existing approaches has materialized in a model which can be used to solve computer vision and graphics problems.

What is lacking is a unified model capable of generating different garment styles, and animating them on any body shape in any pose, while retaining wrinkle detail. To that end, we introduce *TailorNet*, a mixture of Neural Networks (NN) learned from physics based simulations, which decomposes clothing deformations into style, shape and pose – this effectively approximates the physical clothing deformation allowing intuitive control of synthesized animations. In the same spirit as SMPL [32] for bodies, TailorNet learns deformations as a displacements to a garment template in a canonical pose, while the articulated motion is driven by skinning. TailorNet can either take a real garment as input, or generate it from scratch, and drape it on top of the SMPL body for any shape and pose. In contrast to [17], our model predicts how the garment would fit in reality, *e.g*., a medium size garment is predicted tight on a large body, and loose on a thin body.

Learning TailorNet required addressing several technical challenges. To generate different garment styles in a static pose, we compute a PCA subspace using the publicly available digital wardrobe of *real static garments* [6]. To generate more style variation while satisfying garment-human physical constraints, we sample from the PCA subspace, run PBS for each sample, and recompute PCA again, to obtain a *static style subspace*. Samples from this subspace produce variation in sleeve length, size and fit in a static pose. To learn deformation as a function of pose and shape, we generated a semi-real dataset by animating garments (real or samples from the static style subspace) using PBS on top of SMPL [32] body for static SMPL poses, and for different shapes.

Our first observation is that, for a *fixed style* (garment instance) and body shape, predicting high frequency clothing deformations as a function of pose is possible – perhaps surprisingly, our experiments show that, for this task, a simple multi-layer perceptron model (MLP) performs as well as or better than Graph Neural Networks [31, 28] and Image-decoder on a UV-space [29]. In stark contrast, straightforward prediction of deformation as a function of style, shape and pose results in overly smooth un-realistic results. We hypothesize that any attempt to combine training examples smoothes out *high frequency* components, which explains why previous models [46, 17, 18], even for a single style, lack fine scale wrinkles and folds.

These key observations motivate the design of TailorNet: we predict the clothing low frequency geometry with a simple MLP. High frequency geometry is predicted with a mixture of high frequency style-shape specific models, where each specific model consists of a MLP which predicts deformation as a function of pose, and the weights of the mixture are obtained using a kernel which evaluates similarity in style and shape. A kernel with a very narrow bandwidth, prevents smoothing out fine scale wrinkles. Several experiments demonstrate that our model generalizes well to novel poses, predicts garment fit dependent on body shape, retains wrinkle detail, can produce style variations for a garment category (*e.g*., for T-shirts it produces different sizes, sleeve lengths and fit type), and importantly is easy to implement and control. To summarize, the main contributions of our work are:

- The first joint model of clothing style, pose and shape variation, which is simple, easy to deploy and fully differentiable for easy integration with deep learning.
- A simple yet effective decomposition of mesh deformations into low and high-frequency components, which coupled with a mixture model, allows to retain high-frequency wrinkles.
- A comparison of different methods (MLP, Graph Neural Nets and image-to-image translation on UV-space) to predict pose dependent clothing deformation.
- To stimulate further research, we make available a Dataset of 20 aligned real static garments, simulated in 1782 poses, for 9 body shapes, totaling 55800 frames.

Several experiments show that our model generalizes to completely new poses of the AMASS dataset (even though we did not use AMASS to train our model), and produces variations due to pose, shape and style, while being more detailed than previous style specific models [46, 18]. Furthermore, despite being trained on static poses, TailorNet produces smooth continuous animations.

## 2. Related Work

There are two main approaches to animation of clothing: physics based simulation (PBS), and efficient data-driven models learned from offline PBS, or real captures.

**Physics Based Simulation (PBS).** Super realistic animations require simulating millions of triangles [53, 47, 22], which is computationally expensive. Many works focus on making simulation efficient [16] by adding wrinkles to low-resolution simulations [15, 25, 26, 55], or using simpler mass-spring models [43] and position based dynamics [38, 37] compromising accuracy and physical correctness for speed. Tuning the simulation parameters is a tedious task that can take weeks. Hence, several authors attempted to infer physical parameters mechanically [35, 57], from multi-view video [45, 63, 50], or from perceptual judgments of humans [49], but they only work in controlled settings and still require running PBS. PBS approaches typically require designing garments in 2D, grading them, adjusting them to the 3D body, and fine tuning parameters, which can take hours if not weeks, even for trained experts.

**Data-driven cloth models.** One way to achieve realism is to capture real clothing on humans from images [6, 5, 3, 4, 19], dynamic scans [41, 39] or RGBD [52, 51] and retarget it to novel shapes, but this is limited to re-animating the same motion on a novel shape. While learning pose-dependent models from real captures [62, 29, 33] is definitely an exciting route, accurately capturing sufficient real data is still a major challenge. Hence, these models [62, 29, 33] only demonstrate generalization to motions similar to the training data.

Another option is to generate data with PBS, and learn efficient data-driven models. Early methods relied on linear auto-regression or efficient nearest neighbor search to predict pose [10, 26, 56], or pose and shape dependent clothing deformation [17]. Recent ones are based on deep learning, and vary according to the factor modeled (style or pose and shape), the data representation, and architecture used. Style variation is predicted from a user sketch with a VAE [58] but the pose is fixed. For a single style, pose and shape variation is regressed with MLPs and RNNs [46, 62], or Graph-NNs that process body and garment [18]. Pose effects are predicted with a normal map [29] or a displacement map [23, 60] in UV-space. These models [10, 18, 17] tend to produce over-smooth results, with the exception of [29], but the model is trained for a single garment and subject, and as mentioned before generalization to in the wild motions (for example CMU [11]) is not demonstrated. Since there is no consensus on what representation and learning model is best suited for this task, for a fixed style and shape, we compare representations and architectures, and find that MLPs perform as well as more sophisticated Graph-NNs or image-to-image translation on UV-space. While we draw inspiration from previous works, unlike our model, none of them can jointly model style, pose and shape variation.

**Pixel based models.** An alternative to 3D cloth modeling is to retrieve and warp images and videos [61, 9]. Recent methods use deep learning to generate pixels [20, 14, 13, 64, 66, 68, 21, 44, 54, 67], often guided by 2D warps, or smooth 3D models [30, 65, 59, 48], or learn to transfer texture from cloth images to 3D models [36]. They produce (at best) photo-realistic images in controlled settings, but do not capture the 3D shape of wrinkles, cannot easily control motion, view-point, illumination, shape and style.

## 3. Garment Model Aligned with SMPL

Our base garment template is aligned with SMPL [32] as done in [41, 6]. SMPL represents the human body $M(\cdot)$ as a parametric function of pose($\boldsymbol{\theta}$) and shape($\boldsymbol{\beta}$)

$$M(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(T(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}) \tag{1}$$

$$T(\boldsymbol{\beta}, \boldsymbol{\theta}) = \mathbf{T} + B_s(\boldsymbol{\beta}) + B_p(\boldsymbol{\theta}). \tag{2}$$

composed of a linear function $T(\boldsymbol{\beta}, \boldsymbol{\theta})$ which adds displacements to base mesh vertices $\mathbf{T} \in \mathbb{R}^{n \times 3}$ in a T-pose, followed by learned skinning $W(\cdot)$. Specifically, $B_p(\cdot)$ adds pose-dependent deformations, and $B_s(\cdot)$ adds shape dependent deformations. $\mathbf{W}$ are the blend weights of a skeleton $J(\cdot)$.

We obtain the garment template *topology* as a submesh (with vertices $\mathbf{T}^G \in \mathbb{R}^{m \times 3}$) of the SMPL template, with vertices $\mathbf{T}$. Formally, the indicator matrix $\mathbf{I} \in \mathbb{Z}^{m \times n}$ evaluates to $\mathbf{I}_{i,j} = 1$ if garment vertex $i \in \{1 \ldots m\}$ is associated with body shape vertex $j \in \{1 \ldots n\}$. The particular garment *style* draped over $M(\boldsymbol{\beta}, \boldsymbol{\theta})$ in a *0-pose* is encoded as displacements $\mathbf{D}$ over the unposed body shape $T(\boldsymbol{\theta}, \boldsymbol{\beta})$. Since the garment is associated with the underlying SMPL body, previous works [6, 41] deform every clothing vertex with its associated SMPL body vertex function. For a given style $\mathbf{D}$, shape $\boldsymbol{\beta}$ and pose $\boldsymbol{\theta}$, they deform clothing using the un-posed SMPL function $T\boldsymbol{\theta}, \boldsymbol{\beta})$

$$T^G(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{D}) = \mathbf{I}\, T(\boldsymbol{\beta}, \boldsymbol{\theta}) + \mathbf{D}, \tag{3}$$

followed by the SMPL skinning function in Eq. 1

$$G(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{D}) = W(T^G(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{D}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}). \tag{4}$$

Since $\mathbf{D}$ is *fixed*, this assumes that *clothing deforms in the same way as the body*, which is a practical but clearly *over-simplifying assumption*. Like previous work, we also decompose deformation as a non-rigid component (Eq. 3) and an articulated component (Eq. 3), but unlike previous work, we learn non-rigid deformation $\mathbf{D}$ as a function of pose, style and shape. That is, we learn true clothing variations as we explain in the next section.

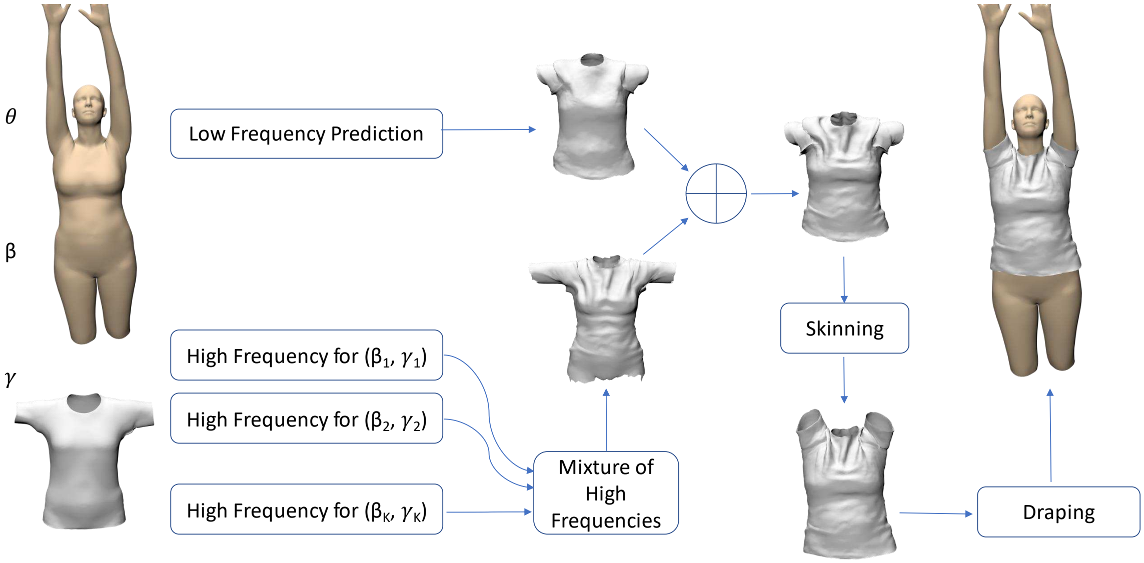
θ
β
γ
Low Frequency Prediction
High Frequency for (β1, γ1)
High Frequency for (β2, γ2)
High Frequency for (βK, γK)
Mixture of High Frequencies
Skinning
Draping

Figure 2. Overview of our model to predict the draped garment with style $\boldsymbol{\gamma}$ on the body with pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$. Low frequency of the deformations are predicted using a single model. High frequency of pose dependent deformations for $K$ prototype shape-style pairs are separately computed and mixed using a RBF kernel to get the final high frequency of the deformations. The low and high frequency predictions are added to get the unposed garment output, which is posed to using standard skinning to get the garment.

| Method | Static/Dynamic | Pose Variations | Shape Variations | Style Variations | Model Public | Dataset Public |
|---|---|---|---|---|---|---|
| Santesteban *et al.* [46] | Dynamic | ✓ | ✓ | ✗ | ✗ | ✗ |
| Wang *et al.* [58] | Static | ✗ | ✓ | ✓ | ✓ | ✓ |
| DeepWrinkles [29] | Dynamic | ✓ | ✓ | ✗ | ✗ | ✗ |
| DRAPE [17] | Dynamic | ✓ | ✓ | ✗ | ✗ | ✗ |
| GarNet [18] | Static | ✓ | ✓ | ✗ | ✗ | ✓ |
| Ours | Static | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1. Comparison of our method with other works. Ours is the first method to model the garment deformations as a function of pose, shape and style. We also make our model and dataset public for further research.

# 4. Method

In this section, we describe our decomposition of clothing as non-rigid deformation (due to pose, shape and style) and articulated deformation, which we refer to as *un-posing* (Section 4.1). The first component of our model is a subspace of garment styles which generates variation in A-pose (Section 4.2). As explained in the introduction, pose-models specific to a *fixed shape and style* (explained in Section 4.3), do preserve high-frequencies, but models that combine different styles and shapes produce overly smooth results. Hence, at the heart of our technique is a model, which predicts low frequency with a straight-forward MLP, and high-frequency with a mixture model Sections 4.2–4.4. An overview of our method is shown in Fig. 2.

## 4.1. Un-posing Garment Deformation

Given a set of *simulated* garments $\mathbf{G}$ for a given pose $\boldsymbol{\theta}$ and shape $\boldsymbol{\beta}$, we first disentangle non-rigid deformation from articulation by *un-posing* – we invert the skinning $W(\cdot)$ function

$$\mathbf{D} = W^{-1}(\mathbf{G}, J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W}) - \mathbf{I}\, T(\boldsymbol{\beta}, \boldsymbol{\theta}), \tag{5}$$

and subtract the body shape in a canonical pose, obtaining un-posed non-rigid deformation $\mathbf{D}$ as displacements from the body. Since joints $J(\cdot)$ are known, computing $W^{-1}(\cdot)$ in Eq. 5 entails un-posing every vertex $\hat{\mathbf{v}}_j = (\sum_k \mathbf{W}_{k,j}\mathbf{H}_k(\boldsymbol{\theta}, J(\boldsymbol{\beta})))^{-1}\mathbf{v}_j$, where $\mathbf{H}_k(\boldsymbol{\theta}, J(\boldsymbol{\beta})) \in SE(3)$ are the part transformation matrices, and $\hat{\mathbf{v}}_j \in \mathbb{R}^3$ is the un-posed vertex. Non-rigid deformation $\mathbf{D}$ in the unposed space is affected by body pose, shape and the garment style (size, sleeve length, fit). Hence, we propose to learn deformation $\mathbf{D}$ as a function of shape $\boldsymbol{\beta}$, pose $\boldsymbol{\theta}$ and style $\boldsymbol{\gamma}$, i.e. $D(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}) : \mathbb{R}^{|\boldsymbol{\theta}|} \times \mathbb{R}^{|\boldsymbol{\beta}|} \times \mathbb{R}^{|\boldsymbol{\gamma}|} \mapsto \mathbb{R}^{m\times 3}$. The model should be realistic, easy to control and differentiable.

## 4.2. Generating Parametric Model of Style

We generate style variation $\boldsymbol{\gamma}$ in A-pose by computing a PCA subspace using the public 3D garments of [6]. Although the method of Bhatnagar *et al*. [6] allows to transfer the 3D garment to a body shape in a canonical pose and shape ($\{G(\boldsymbol{\theta}_0, \boldsymbol{\beta}_0, \mathbf{D}_i)\}$), physical constraints might be violated – garments are sometimes slightly flying on top of the body, and wrinkles do not correspond to an A-pose. In order to generate style variation while satisfying physics constraints, we alternate sampling form the PCA space and running PBS on the samples. We already find good results alternating PBS and PCA two times, to obtain a sub-space of style variation in a *static A-pose* – with PCA coefficients $\boldsymbol{\gamma}$. Fig. 3 shows the parametric style space for t-shirt. This is however, a model in a fixed A-pose and shape. Different garment *styles will deform differently* as a function of *pose* and *shape*. In Section 4.3 we explain our style-shape specific model of pose, and in Section 4.4 we describe our joint mixture model for varied shapes and styles.

### 4.3. Single Style-Shape Model

For a fixed garment style $\boldsymbol{\gamma}$ and a fixed body shape $\boldsymbol{\beta}$ pair, denoted as $\phi = (\boldsymbol{\gamma}, \boldsymbol{\beta}) \in \mathcal{X}$, we take paired simulated data of body poses and garment displacements $\{(\boldsymbol{\theta}_i, \mathbf{D}_i^{\phi})\}$ ,computed according to Eq. 5, and train a model $D_\phi : \mathbb{R}^{|\boldsymbol{\theta}|} \rightarrow \mathbb{R}^{m \times 3}$ to predict pose dependent displacements. In particular, we train $D_\phi$ using a multi-layer perceptron(MLP) to minimize the L1-loss between the predicted displacements $D_\phi(\boldsymbol{\theta}_i)$ and the ground-truth displacements $\mathbf{D}_i^{\phi}$. We observe that $D_\phi$ predicts reasonable garment fit along with fine-scale wrinkles and generalizes well to unseen poses, but this model is *specific to a particular shape and style.*

### 4.4. TailorNet

For the sake of simplicity, it is tempting to directly regress all non-rigid clothing deformations as function of pose, shape and style jointly, *i.e*., learning $D(\boldsymbol{\theta}, \phi) : \mathbb{R}^{|\boldsymbol{\theta}|} \times \mathcal{X} \mapsto \mathbb{R}^{m \times 3}$ directly with an MLP. However, our experiments show that, while this produces accurate predictions quantitatively, qualitatively results are overly *smooth lacking realism*. We hypothesize that any attempt to combine geometry of many samples with varying high frequency (fine wrinkles) smoothes out the details. This might be a potential explanation for the smooth results obtained in related works [46, 18, 17] – even for single style models. Our idea is to decompose garment mesh vertices, in an unposed space $\hat{\mathbf{G}} = W^{-1}(\mathbf{G}, J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathbf{W})$, into a smooth low-frequency shape $\hat{\mathbf{G}}^{LF}$, and a high frequency shape $\hat{\mathbf{G}}^{HF}$ with diffusion flow. Let $f(\mathbf{x}, t) : \mathcal{G} \mapsto \mathbb{R}$ be a function on the garment surface, then it is smoothed with the diffusion equation:

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = \lambda \Delta f(\mathbf{x}, t) \qquad (6)$$

which states that the function changes over time by a scalar diffusion coefficient $\lambda$ times its spatial Laplacian $\Delta f$. In order to smooth mesh geometry, we apply the diffusion equation to the vertex coordinates $\mathbf{g}_i \in \hat{\mathbf{G}}$ (which are interpreted as a discretized function on the surface)

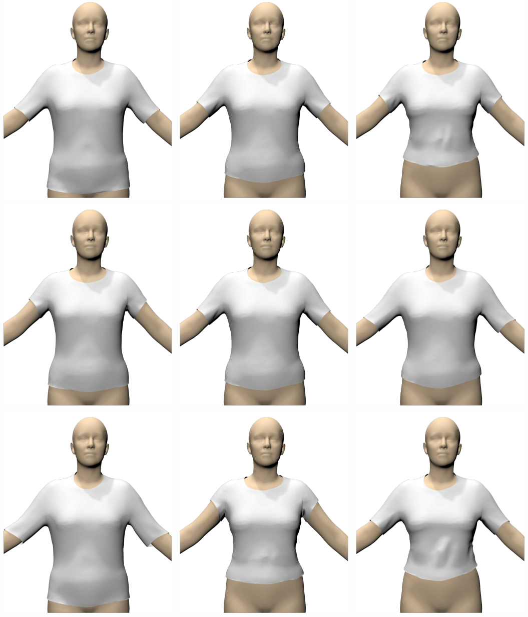

Figure 3. Overview of T-shirt style space. First component(top) and second component(middle) changes the overall size and the sleeve length respectively. Sampling from this style space generates a wide range of T-shirt styles (bottom). For the first two rows, corresponding components are $-1.5\sigma, 0, 1.5\sigma$ from left to right.

$$\mathbf{g}_i = \mathbf{g}_i + \lambda \Delta \mathbf{g}_i, \qquad (7)$$

where $\Delta \mathbf{g}_i$ is the discrete Laplace-Beltrami operator applied at vertex $\mathbf{g}_i$, and $\lambda$ and the number of iterations control the level of smoothing. Eq. 7 is also known as Laplacian smoothing [12, 7]. We use $\lambda = 0.15$ and 80 iterations, to obtain a smooth low-frequency mesh $\hat{\mathbf{G}}^{LF}$ and a high-frequency residual $\hat{\mathbf{G}}^{HF} = \hat{\mathbf{G}} - \hat{\mathbf{G}}^{LF}$. We then subtract the body shape $\mathbf{I}T(\boldsymbol{\beta}, \boldsymbol{\theta})$ as in Eq. 5, and predict displacement components separately as

$$D(\boldsymbol{\theta}, \phi) = D^{LF}(\boldsymbol{\theta}, \phi) + \sum_{k=1}^{K} \Psi(\phi, \phi_k) D_{\phi,k}^{HF}(\boldsymbol{\theta}), \qquad (8)$$

where the low-frequency component is predicted with an MLP $D^{LF}(\boldsymbol{\theta}, \phi)$ (smooth but accurate), whereas the high frequency component is predicted with a mixture of style-shape $\phi = (\boldsymbol{\gamma}, \boldsymbol{\beta})$ specific models of pose $D_\phi^{HF}(\boldsymbol{\theta})$. As we show in the experiments, style-shape $\phi$ specific high frequency models retain details. We generalize to new shape-styles beyond the prototypes $D_{\phi_k}^{HF}(\boldsymbol{\theta})$ with a convex combination of specific models. The mixture weights are computed with a kernel $\Psi(\phi_1, \phi_2) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ with a narrow bandwidth to *combine only similar wrinkle patterns*

$$\Psi(\phi, \phi_k) = \exp\left(-\frac{\text{dist}\,(g(\phi), g(\phi_k))}{\sigma_x}\right), \qquad (9)$$

where $g(\phi) : \mathcal{X} \mapsto \mathbb{R}^{m \times 3}$ is a shallow MLP which maps from style-shape to the garment displacements $\mathbf{D}$ in a canonical A-pose. Ideally, the kernel should measure similarity in style and shape, but this would require simulating training data for every possible pose-shape-style combination, which is inefficient and resource intensive. Our key simplifying assumption is that two garments on two different people will deform similarly if their displacements $g(\phi) = \mathbf{D}$ to their respective bodies is similar – this measures clothing fit similarity. While this is an approximation, it works well in practice.

The bandwidth $\sigma_x$ is a free-parameter of our model, allowing to generate varying high-frequency detail. We find qualitatively good results by keeping $\sigma_x$ small in order to combine only nearby samples. We postprocess the output to remove the garment intersections with the body.

**Choosing K Style-Shape Prototypes** For each chosen style-shape pair, we simulate a wide variety of poses, which is time consuming. Hence, we find $K$ prototype shape-styles $\phi$ such that they cover the space of static displacements in A-pose $D(\boldsymbol{\beta}, \boldsymbol{\theta}_0, \boldsymbol{\gamma})$. Specifically, we want to choose $K$ style-shape pairs such that any other shape-styles can be approximated as a convex combination of prototypes with least error – this is a non-convex problem with no global optimum guarantees. While iterative methods like K-SVD [2] might yield better coverage, here we use a simple but effective greedy approach to choose good prototypes.

We take a dataset of $X$ garments $\{D(\boldsymbol{\theta}_0, \phi)\}_{i=1 \ldots X}$ in different styles $\boldsymbol{\gamma}_i$ draped on different body shapes $\boldsymbol{\beta}_i$ in canonical pose. We start with a pool of 9 body shapes in one common style and try to fit each of the other style-shape pairs as a convex combination of this pool. Then we take the style-shape with the highest approximation error and add it to the pool. We repeat this process until we get the pool of $K$ style-shape pairs. We find all shape-styles can be well approximated when $K \geqslant 20$. Thus, we use $K = 20$ here.

# 5. Dataset

We learn our model from paired data of parameters and the garment which we obtained from simulation. We use a commercial 3D cloth design and simulation tool Marvelous Designer [1] to simulate these garments. Instead of designing and positioning garments manually, we use the publicly available digital wardrobe [6] captured from the real world data which includes variations in styles, shapes and poses. Although we describe our dataset generation with reference to T-shirt, method remains the same for all garments.

## 5.1. Garments to Generate Styles

We first retarget [41] the 3D garments from the digital wardrobe [6] to the canonical shape($\boldsymbol{\beta}_0$) and slowly simulate them to canonical pose($\boldsymbol{\theta}_0$). We get the dataset $\{G(\boldsymbol{\beta}_0, \boldsymbol{\theta}_0, \mathbf{D}_i)\}$ of 43 garments to learn style-space as in Section 4.2. However, the learnt style-space may not be intuitive due to limited variation in the data, and may contain irregular patterns and distortions owing to the registration process. So we sample 300 styles from this PCA model, simulate them again to get a larger dataset with less distortions. With 2 iterations of PCA and simulation, we generate consistent and meaningful style variations. We find first 2 PCA components enough to represent $\boldsymbol{\gamma}$ parameters. See Figure 3.

## 5.2. Shape, Pose and Style Variations

We choose 9 shapes manually as follows. We sample the first two shape components $\beta$ at 4 equally spaced intervals, while leaving the other components to zero. To these $4 \times 2 = 8$ shapes, we add canonical zero shape $\boldsymbol{\beta}_0$. We choose 25 styles in a similar way - by sampling the first two sytle $\gamma$ components.

We simulate all combinations of shapes and styles in canonical pose, which results in $25 \times 9 = 225$ instances. We choose 20 prototypes out of 225 style-shape pairs as training style-shape pairs using the approach mentioned in Section 4.4. We randomly choose 20 more style-shape pairs for testing.

## 5.3. Simulation details

For pose variations, we use 1782 static SMPL poses, including a wide range of poses, including extreme ones. For a given style-shape $\phi_k$ and poses $\{\boldsymbol{\theta}_i\}$, we simulate them in one sequence. The body starts with canonical pose and greedily transitions to the nearest pose until all poses are traversed. To avoid dynamic effects (in this paper we are interested in quasi static deformation), we insert linearly interpolated intermediate poses, and let the garment relax for few frames. PBS simulation fails sometimes, and hence we remove frames with self-interpenetration from the dataset.

For training style-shape pairs, we simulate SMPL poses with interpolated poses. For testing style-shape pairs, we simulate a subset of SMPL poses with a mix of seen and unseen poses. Finally, we get 4 splits as follows: (1) train-train : 20 training style-shape pairs each with 2600 training poses. (2) train-test : 20 training style-shape pairs each with 179 unseen poses. (3) test-train : 20 testing style-shape pairs each with 35 training poses. (4) test-test : 20 testing style-shape pairs each with 131 unseen poses.

# 6. Experiments

We evaluate our method on T-Shirt quantitatively and qualitatively and compare it to our baseline, and previous works. Our baseline and TailorNet use several MLPs - each of them has two hidden layers with ReLU activation and a dropout layer. We arrive to optimal hyperparameters by tuning our baseline, and then keep them constant to train all other MLPs. See suppl. for more details.

### 6.1. Results of Single Style-Shape Model

We train single style-shape models $D_{\phi}^{HF}(\boldsymbol{\theta})$ on $K$ style-shape pairs separately. The mean per vertex test error of each model varies depending upon the difficulty of particular style-shape. The average error across all $K$ models is $8.04$ mm with maximum error of $14.50$ mm for a loose fitting, and minimum error of $6.56$ mm for a fit fitting.

For single style-shape model, in the initial stages of the work, we experimented with different modalities of 3D mesh learning: UV map, similar to [29], and graph convolutions (Graph CNN) on the garment mesh. For learning UV map, we train a decoder from the input pose parameters to UV map of the garment displacements. For Graph CNN [27] by putting the input $\boldsymbol{\theta}$ on each node of the graph input. We report results on two style-shape pairs in Table 2, which shows that a simple MLP is sufficient to learn pose dependent deformations for a fixed style-shape $\phi$.

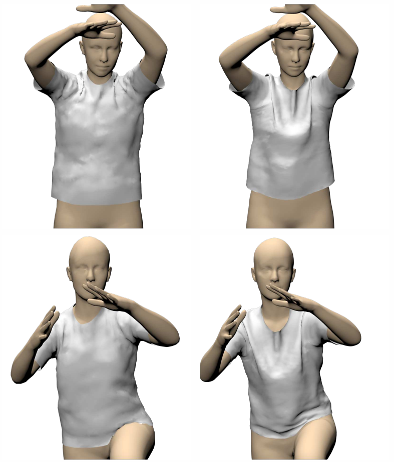

Figure 4. Results on unseen pose, shape and style. Prediction by [46] (left), our mixture model (right). Note that our mixture model is able to retain more meaningful folds and details on the garment.

| Style-shape | MLP | UV Decoder | Graph CNN |
|---|---|---|---|
| Loose-fit | 14.5 | 15.9 | 16.1 |
| Tight-fit | 10.1 | 11.4 | 11.7 |

Table 2. Mean per vertex error in mm for the pose dependent prediction by MLP, UV Decoder and Graph CNN for two style-shapes.

### 6.2. Results of TailorNet

We define our baseline $f_{BL}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}) : \mathbb{R}^{|\boldsymbol{\theta}|+|\boldsymbol{\beta}|+|\boldsymbol{\gamma}|} \rightarrow \mathbb{R}^{m\times 3}$ implemented as a MLP to predict the displacements. Table 3 shows that our mixture model outperforms the baseline by a slight margin on 3 testing splits.

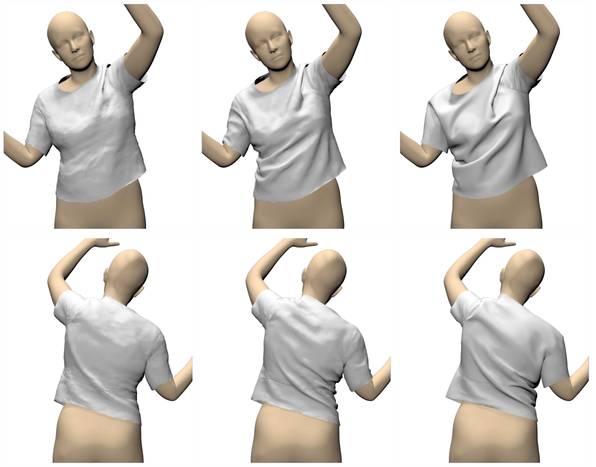

Figure 5. Baseline method (left) smooths out the fine details over the garment. TailorNet (middle) is able to retain as many wrinkles as PBS groundtruth (right).

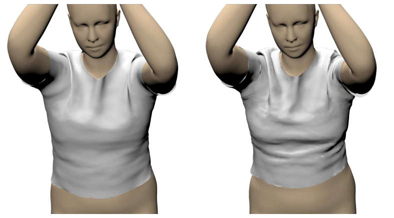

Figure 6. (left) Mixing the outputs of prototypes directly without decomposition smooths out the fine details. (right) The decomposition into high and low frequencies allows us to predict good garment fit with details.

| Split No. | Style-shape set | Pose set | Our Baseline | Our Mixture Model |
|---|---|---|---|---|
| 2 | train | test | 10.6 | 10.2 |
| 3 | test | train | 11.7 | 11.4 |
| 4 | test | test | 11.6 | 11.4 |

Table 3. Mean per vertex error in mm for 3 testing splits in Section 5.3. Our mixture performs slightly better than the baseline quantitatively, and significantly better qualitatively.

Qualitatively, TailorNet outperforms the baseline and previous works. Figure 4 shows the garment predicted by Santesteban *et al*. [46] and our mixture model. Figure 5 shows the qualitative difference between the output by TailorNet and our baseline trained on the same dataset.

To validate our choice to decompose high and low frequency details, we consider a mixture model where individual MLPs predict style-shape dependent displacements directly without decomposition. Figure 6 shows that although it can approximate the overall shape well, it looses the fine wrinkles. TailorNet retains the fine details and provides intuitive control over high frequency details.

**Sequences of AMASS [34]:** TailorNet generalizes (some poses shown in Fig. 7) despite being trained on com-

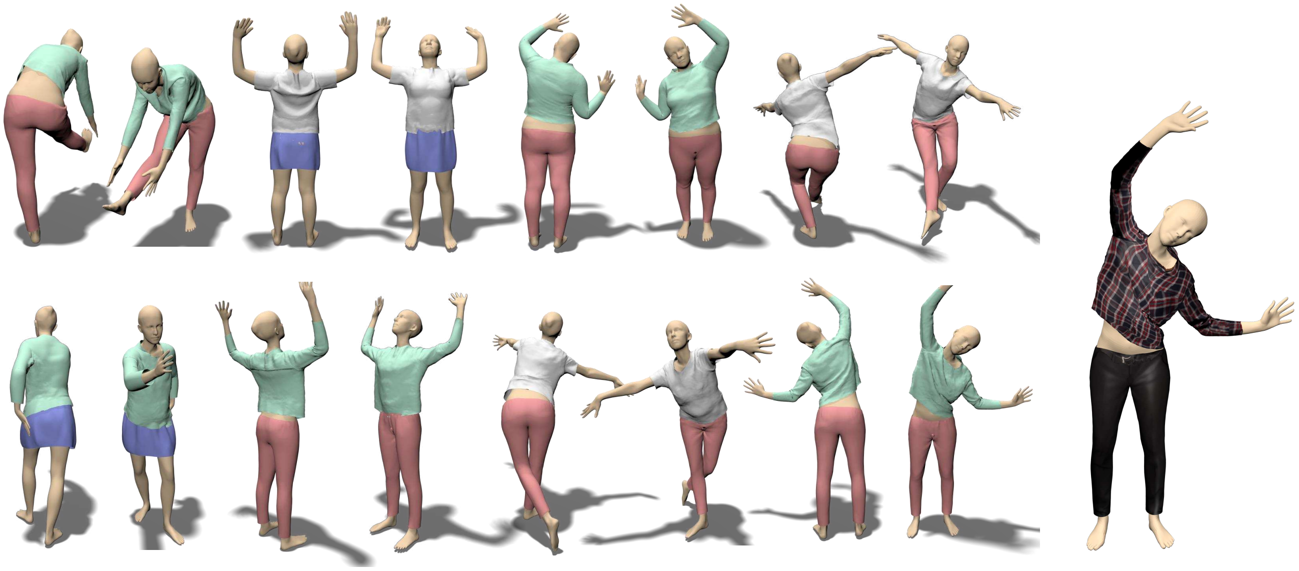

Figure 7. Left: Predictions of TailorNet for multiple garments and completely unseen poses. Right: TailorNet prediction with a real texture.

pletely different poses. Notably, the results are temporally coherent, despite not modelling the dynamics. See suppl. video for visualization.

### 6.3. Multiple Garments

To show the generalizability of TailorNet, we trained it separately for 3 more garments - Shirt, Pants and Skirt. Since skirt do not follow the topology of template body mesh, we attach a skirt template to the root joint of SMPL [41]. For each of these garments, we simulate the dataset for 9 shapes with a common style and trained the model. Figure 7 shows the the detailed predictions by TailorNet. Since we use the base garments, which come from a *real* digital wardrobe [6], we can also transfer the textures from real scan on our predictions.

### 6.4. Runtime Performance

We implement TailorNet using PyTorch [40]. On a gaming laptop with an NVIDIA GeForce GTX 1060 GPU and Intel i7 CPU, our approach runs from 1 to 2 ms per frame, which is 1000 times faster than PBS it is trained from. Just on CPU, it runs 100 times faster than PBS.

## 7. Discussion and Conclusion

TailorNet is the first data-driven clothing model of pose, shape and *style*. The experiments show that a simple MLP approximates clothing deformations with an accuracy of 11 mm, which is as good as more sophisticated graph-CNN, or displacements prediction in UV-space, but shares the same limitations with existing methods: when it is trained with different body shapes (and styles in our case) results are overly smooth and lack realism. To address this, we proposed a model which predicts low and high frequencies separately. Several experiments show that our narrow bandwidth mixture model to predict high-frequency preserves significantly more detail than existing models, despite having a much harder task, that is, modelling pose dependent deformation as a function of shape and style jointly.

In future work, we plan to fit the model to scans, images and videos, and will investigate refining it using video data in a self-supervised fashion, or using real 3D cloth captures [41]. We also plan to make our model sensitive to the physical properties of the cloth fabric and human soft-tissue [42].

Our TailorNet 1000 times faster than PBS, allows easy control over style, shape and pose without manual editing. Furthermore, the model is differentiable, which is crucial for computer vision and learning applications. We will make TailorNet and our dataset publicly available, and will continuously add more garment categories and styles to make itl even more widely useable. TailorNet fills a key missing component in existing body models like SMPL and ADAM [32, 24]: realistic clothing, which is necessary for animation of virtual humans, and to explain the complexities of dressed people in scans, images, and video.

**Acknowledgements.** This work is partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans) and Google Faculty Research Award. We thank Bharat Lal Bhatnagar and Garvita Tiwari for insightful discussions, and Dan Casas for providing us test results of [46].

## References

[1] https://www.marvelousdesigner.com/. 6

[2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006. 6

[3] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[4] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018. 3

[5] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 3

[6] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 2, 3, 4, 5, 6, 8

[7] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*. A K Peters, 2010. 5

[8] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4d video textures for interactive character appearance. *Computer Graphics Forum*, 33(2):371–380, 2014. 2

[9] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4D Video Textures for Interactive Character Appearance. *Computer Graphics Forum (Proceedings of EUROGRAPHICS)*, 33(2):371–380, 2014. 3

[10] Edilson de Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K. Hodgins. Stable spaces for real-time clothing. *ACM Trans. Graph.*, 29(4):106:1–106:9, 2010. 2, 3

[11] Fernando De la Torre, Jessica Hodgins, Javier Montano, Sergio Valcarcel, R Forcada, and J Macey. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. *Robotics Institute, Carnegie Mellon University*, 5, 2009. 3

[12] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999. 5

[13] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[14] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. Fw-gan: Flow-navigated warping gan for video virtual try-on. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[15] Russell Gillette, Craig Peters, Nicholas Vining, Essex Edwards, and Alla Sheffer. Real-time dynamic wrinkling of coarse animated cloth. In *Proc. Symposium on Computer Animation*, 2015. 3

[16] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):to appear, 2007. 3

[17] Peng Guan, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black. DRAPE: dressing any person. *ACM Trans. Graph.*, 31(4):35:1–35:10, 2012. 2, 3, 4, 5

[18] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. *CoRR*, abs/1811.10983, 2018. 2, 3, 4, 5

[19] Marc Habermann, Weipeng Xu, , Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *Transactions on Graphics (ToG) 2019*, oct 2019. 3

[20] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *CVPR*, 2018. 3

[21] Wei-Lin Hsiao, Isay Katsman, Chao-Yuan Wu, Devi Parikh, and Kristen Grauman. Fashion++: Minimal edits for outfit improvement. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[22] Chenfanfu Jiang, Theodore Gast, and Joseph Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics (TOG)*, 36(4):152, 2017. 3

[23] Ning Jin, Yilin Zhu, Zhenglin Geng, and Ronald Fedkiw. A pixel-based framework for data-driven clothing. *CoRR*, abs/1812.01677, 2018. 3

[24] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8320–8329, 2018. 8

[25] Ladislav Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph.*, 30(4):93:1–93:10, July 2011. 3

[26] Doyub Kim, Woojong Koh, Rahul Narain, Kayvon Fatahalian, Adrien Treuille, and James F. O'Brien. Near-exhaustive precomputation of secondary cloth effects. *ACM Transactions on Graphics*, 32(4):87:1–7, July 2013. Proceedings of ACM SIGGRAPH 2013, Anaheim. 3

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 7

[28] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 2

[29] Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018. 2, 3, 4, 7

[30] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model of people in clothing. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, Piscataway, NJ, USA, oct 2017. IEEE. 3

[31] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. *CVPR*, 2018. 2

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 3, 8

[33] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael Black. Learning to dress 3d people in generative clothing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 3

[34] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019. 1, 7

[35] E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. Data-driven estimation of cloth simulation models. *Comput. Graph. Forum*, 31(2pt2):519–528, May 2012. 3

[36] Aymen Mir, Thiemo Alldieck, and Gerard Pons-Moll. Learning to transfer texture from clothing images to 3d humans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 3

[37] Matthias Müller. Hierarchical position based dynamics. 2008. 3

[38] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. 3

[39] Alexandros Neophytou and Adrian Hilton. A layered model of human body and garment deformation. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 171–178. IEEE, 2014. 3

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 8

[41] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael Black. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 36(4), 2017. 3, 6, 8

[42] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 34(4):120:1–120:14, Aug. 2015. 8

[43] Xavier Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995. 3

[44] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. Swapnet: Image based garment transfer. In *European Conference on Computer Vision*, pages 679–695. Springer, 2018. 3

[45] Bodo Rosenhahn, Uwe Kersting, Katie Powell, Reinhard Klette, Gisela Klette, and Hans-Peter Seidel. A system for articulated tracking incorporating a clothing model. *Machine Vision and Applications*, 18(1):25–40, 2007. 3

[46] Igor Santesteban, Miguel A. Otaduy, and Dan Casas. Learning-based animation of clothing for virtual try-on. *Comput. Graph. Forum*, 38(2):355–366, 2019. 2, 3, 4, 5, 7, 8

[47] Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):339–350, 2009. 3

[48] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019. 3

[49] Leonid Sigal, Moshe Mahler, Spencer Diaz, Kyna McIntosh, Elizabeth Carter, Timothy Richards, and Jessica Hodgins. A perceptual control space for garment simulation. *ACM Trans. Graph.*, 34(4):117:1–117:10, July 2015. 3

[50] Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. *ACM Trans. Graph.*, 29(6):139:1–139:10, Dec. 2010. 3

[51] Yu Tao, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Dai Quionhai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performance with inner body shape from a depth sensor. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, june 2018. 3

[52] Yu Tao, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Dai Quionhai, Gerard Pons-Moll, and Yebin Liu. Simulcap : Single-view human performance capture with cloth simulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2019. 3

[53] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *ACM Siggraph Computer Graphics*, 21(4):205–214, 1987. 3

[54] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, and Liang Lin. Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 589–604, 2018. 3

[55] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F. O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Transactions on Graphics*, 29(4):107:1–8, July 2010. Proceedings of ACM SIGGRAPH 2010, Los Angles, CA. 3

[56] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F O'Brien. Example-based wrinkle synthesis for clothing animation. In *Acm Transactions on Graphics (TOG)*, volume 29, page 107. ACM, 2010. 3

[57] Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: Modeling and measurement. *ACM Transactions on Graphics, Proc. SIGGRAPH*, 30(4):71:1–11, July 2011. 3

[58] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. Learning a shared shape space for multimodal

garment design. *ACM Trans. Graph.*, 37(6):203:1–203:13, 2018. 2, 3, 4

[59] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019. 3

[60] Jane Wu, Yongxu Jin, Zhenglin Geng, Hui Zhou, and Ronald Fedkiw. Recovering geometric information with learned texture perturbations, 2020. 3

[61] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: Creating new human performances from a multi-view video database. *ACM Trans. Graph.*, 30(4):32:1–32:10, July 2011. 3

[62] Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer. Analyzing clothing layer deformation statistics of 3d human motions. In *European Conf. on Computer Vision*, pages 237–253, 2018. 3

[63] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. Physics-inspired garment recovery from a single-view image. *ACM Transactions on Graphics (TOG)*, 37(5):170, 2018. 3

[64] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[65] Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2018. 3

[66] Bo Zhao, Xiao Wu, Zhi-Qi Cheng, Hao Liu, Zequn Jie, and Jiashi Feng. Multi-view image generation from a single-view. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 383–391. ACM, 2018. 3

[67] Shizhan Zhu, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Chen Change Loy. Be your own prada: Fashion synthesis with structural coherence. In *ICCV*, 2017. 3

[68] Shizhan Zhu, Raquel Urtasun, Sanja Fidler, Dahua Lin, and Chen Change Loy. Be your own prada: Fashion synthesis with structural coherence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1680–1688, 2017. 3