DeePSD: Automatic Deep Skinning And Pose Space Deformation For 3D Garment Animation

Hugo Bertiche[1,2], Meysam Madadi[1,2], Emilio Tylson[1] and Sergio Escalera[1,2]
[1]Universitat de Barcelona and [2]Computer Vision Center, Spain
hugo_bertiche@hotmail.com

Abstract

We present a novel solution to the garment animation problem through deep learning. Our contribution allows animating any template outfit with arbitrary topology and geometric complexity. Recent works develop models for garment edition, resizing and animation at the same time by leveraging the support body model (encoding garments as body homotopies). This leads to complex engineering solutions that suffer from scalability, applicability and compatibility. By limiting our scope to garment animation only, we are able to propose a simple model that can animate any outfit, independently of its topology, vertex order or connectivity. Our proposed architecture maps outfits to animated 3D models into the standard format for 3D animation (blend weights and blend shapes matrices), automatically providing of compatibility with any graphics engine. We also propose a methodology to complement supervised learning with an unsupervised physically based learning that implicitly solves collisions and enhances cloth quality.

1. Introduction

Virtual dressed human animation has been a topic of interest for decades due to its numerous applications in entertainment and videogame industries, and recently, in virtual and augmented reality. Depending on the application we find two main classical computer graphics approaches. On the one hand, Physically Based Simulation (PBS) [6, 17, 23, 24, 29, 30, 33] approaches are able to obtain highly realistic cloth dynamics at the expense of a huge computational cost. On the other hand, Linear Blend Skinning (LBS) [12, 13, 15, 20, 31, 32] and Pose Space Deformation (PSD) [3, 4, 16, 18] models are suitable for environments with limited computational resources or real-time performance demand. To do so, realism is highly compromised. Then, computer graphics approaches present a trade-off between realism and computational performance.

Figure 1: We present a novel approach for outfit animation. Our methodology allows generalization to unseen outfits. It can handle multiple layers of cloth, arbitrary topology and complex geometric details without retraining.

Deep learning has already proven successful in complex 3D tasks [5, 10, 19, 21, 25, 26, 28]. Due to the interest in the topic and the recently available 3D datasets on garments, we see the scientific community pushing this research line [1, 2, 7, 8, 9, 14, 22, 27]. Most proposals are built as non-linear PSD models learnt through deep learning. These methods yield models describing one or few garment types and, therefore, they lack on generalization capabilities. To overcome this, recent works propose encoding garment types as a subset of body vertices [7, 22]. This allows generalizing to more garments, yet bounds its representation capacity to body homotopies only. Thus, these approaches need to model each garment individually and cannot handle details such as pockets nor multiple layers of cloth, heavily hurting their scalability and applicability in real life scenarios.

We propose learning a mapping from the space of template outfits to the space of animated 3D models. We will show how this allows generalization to completely unseen garments with arbitrary topology and vertex connectivity. We can achieve this by identifying edition/resizing and animation as separate tasks, and focusing on the latter. Our method works with whole outfits (instead of single garments), multiple layers of cloth and resolutions, while also allowing complex geometric details (see Fig. 1 for some

examples). Furthermore, we achieve this with a simple and small-sized neural network. The list of our contributions is as follows:

- **Outfit Generalization.** To the best of our knowledge, our proposal is the only work able to animate completely unseen outfits without additional training. This greatly increases applicability in scenarios with ever-growing number of outfits, such as virtual try-ons and videogames, where customization is key.
- **Compatibility.** Our methodology does not predict garment vertex locations, but blend weights and blend shapes matrices. This is the standard on 3D animation, and it is therefore compatible with all graphics engines. Also, it benefits from the exhaustive optimization on animation pipelines. Pose Space Deformations are a specific case of blend shapes that are combined consistently with object pose.
- **Physical Consistency.** Related works require a final post-processing step for collision solving. Alternatively, works that train with a collision-solving loss need to find a compromise between physical constraints and vertex error. Thus, predictions still show collisions. We propose to train an independent model branch such that physical consistency losses and supervised losses do not hinder each other. This yields quasi-collision free and cloth-consistent predictions while leveraging the data as much as possible.
- **Explainability.** Mapping outfits to animated 3D models yields a more intuitive work pipeline for CGI artists. Recent works try to address garment resizing/edition along animation by encoding styles into parametric representations [7, 22]. Thus, expert knowledge is required to obtain the desired results by tuning style parameters.

2. State of the art

Computer Graphics. Obtaining realistic cloth behaviour is possible through **PBS** (Physically Based Simulation), commonly through the well known *mass-spring* model. Literature on the topic is extensive, focused on improving the efficiency and stability of the simulation by simplifying and/or specializing on specific setups [6, 23, 24, 30], or proposing new energy-based algorithms to enhance robustness, realism and generalization to other soft bodies [17]. Other works propose leveraging the parallel computational capabilities of modern GPUs [29, 33]. These approaches achieve high realism at the expense of a great computational cost. Thus, PBS is not an appropriate solution when real-time performance is required or computational capacity is limited (e.g. in portable devices). On the other hand, for applications that prioritize performance, **LBS** (Linear Blend Skinning) is the standard approach on computer graphics for animation of 3D models. Each vertex of the object to animate is attached to a skeleton through a set of blend weights that are used to linearly combine joint transformations. In garment domain, outfits are attached to the skeleton driving body motion. This approach has also been widely studied [12, 13, 15, 20, 31, 32]. While it is possible to achieve real-time performance, cloth dynamics are highly non-linear, which results in a significant loss of realism when applied to garments.

Learning-Based. Due to the drawbacks found in the classical LBS approach, **PSD** (Pose Space Deformation) models appeared [16]. To avoid artifacts due to skinning, corrective deformations are applied to the mesh in rest pose. Additionally, PSD handles pose-dependant high frequency details of 3D objects. While hand-crafted PSD is possible, in practice, it is learnt from data. We find applications of this technique for body models [3, 4, 18], where deformation bases are computed through linear decomposition of registered body scans. Similarly, in garment domain, Guan et al. [9] apply the same techniques for a few template garments on data obtained through simulation. Lähner et al. [14] also propose linearly learnt PSD for garments, but conditioned on temporal features processed by an RNN to achieve a non-linear mapping. Later, Santesteban et al. [27] propose an explicit non-linear mapping for PSD through an MLP for a single template garment. The main drawback of these approaches is that PSD must be learnt for each template garment, which in turns requires new simulations to obtain the corresponding data. To address this issue, many researchers propose an extension of a human body model (SMPL [18]), encoding garments as additional displacements and topology as subsets of vertices [1, 2, 7, 8, 22]. Alldieck et al. [1, 2] propose a single model for body and clothes, first as vertex displacements and later as texture displacement maps, to infer 3D shape from single RGB images. Similarly, Bhatnagar et al. [8] also learn a space for body deformations to encode outfits, plus an additional segmentation to separate body and clothes, also to infer 3D garments from RGB. Jiang et al. [11] propose 3D outfit retrieval from images and predicting the corresponding blend weights w.r.t. SMPL skeleton using, as labels, the weights of the nearest skin vertices. Patel et al. [22] encode a few different garment types as subsets of body vertices and propose a strategy to explicitly deal with high frequency pose-dependant cloth details for different body shapes and garment styles. Bertiche et al. [7] encode thousands of garments on top of the human body by masking its vertices. They learn a continuous space for garment types, on which later they condition, along with the pose, the vertex deformations. Using a body model to represent garments allows handling multiple types with a single model. Nonetheless, it is still limited to single garments, as it cannot work with multiple layers of cloth. For the same reason, they cannot handle complex garment details. This reduces their appli-

cability in real scenarios. Our proposed methodology allows working with arbitrary topologies, number of layers and complex details. Additionally, output format is highly efficient and allows easy integration into graphics engines, increasing compatibility and applicability.

3. Predicting Animated 3D Models

Computer graphics 3D animated models are constructed using skinning and/or blend shapes. In the former, given a 3D mesh with N vertices as $\mathbf{T} \in \mathbb{R}^{N\times 3}$ and a skeleton with K joints as $\mathbf{J} \in \mathbb{R}^{K\times 3}$, each mesh vertex is attached to each joint with a blend weights matrix $\mathbf{W} \in \mathbb{R}^{N\times K}$. Then, animating the 3D mesh can be achieved by posing skeleton $\mathbf{J}$ through linear transformation matrices (rotation, scaling and translation). Vertex transformation matrices are obtained as a weighted average of joint transformations as described by blend weights. For realistic human and garment animation, only rotations are applied to the joints, and thus, an axis-angle representation is used for pose as $\theta \in \mathbb{R}^{K\times 3}$. For the latter, given $\mathbf{T}$ as defined above, a blend shapes matrix as $\mathbf{D} \in \mathbb{R}^{M\times N\times 3}$ encodes M different deformations (shapes) $D_i \in \mathbb{R}^{N\times 3}$ for $\mathbf{T}$. To animate the mesh, M shape keys are required. These keys are used to linearly combine blend shapes to obtain a final deformation for $\mathbf{T}$. Temporal evolution of shape keys animates the mesh. More complex 3D models use a combination of both techniques. First, $\mathbf{T}$ is linearly deformed through blend shapes and later posed along skeleton $\mathbf{J}$ according to blend weights. Whenever shape keys are defined as a function of skeleton pose, we have Pose Space Deformations driven by *pose* keys. More formally, in human and garment animation domain:

$$\mathbf{V}_\theta = W(\mathbf{T} + \sum_i^M f(\theta)_i D_i, \mathbf{J}, \theta, \mathbf{W}) \tag{1}$$

Where $W(\cdot)$ is the skinning function that poses mesh vertices as described by $\mathbf{J}$ and θ, $\mathbf{V}_\theta$ is the posed vertices, $f(\cdot)$ is a function that maps pose θ to M pose keys and D_i are the shapes within blend shapes matrix $\mathbf{D}$. These techniques are the standard for 3D animation. All current graphics engines are compatible with these methods.

An example for this is SMPL [18] (human body model). SMPL consists of a template mesh with vertices $\mathbf{T} \in \mathbb{R}^{6890\times 3}$, an skeleton $\mathbf{J} \in \mathbb{R}^{24\times 3}$, a blend weights matrix $\mathbf{W} \in \mathbb{R}^{6890\times 24}$ and two blend shapes matrices, one to represent different body shapes, $\mathbf{D}_{shape} \in \mathbb{R}^{10\times 6890\times 3}$, and another for Pose Space Deformations, $\mathbf{D}_{PSD} \in \mathbb{R}^{207\times 6890\times 3}$. Body shape is defined by shape keys $\beta \in \mathbb{R}^{10}$ and Pose Space Deformations by pose keys as flattened rotation matrices (removing global orientation) $R \in \mathbb{R}^{207}$. Because of its formulation SMPL is compatible with current graphics engines. Through this paper, we use SMPL as support body model for animating outfits.

In this work we present a novel approach for garment animation. While recent works are already leveraging skinning blend weights w.r.t. body skeleton to drive garment motion, authors usually rely on complex formulations for Pose Space Deformations, hindering their compatibility with graphics engines and reducing significantly their applicability in real scenarios. We propose learning a mapping from template outfit (canonical pose) meshes to their corresponding blend weights and blend shapes matrices through deep learning. That is, learning a neural network $\mathcal{M}$ as:

$$\mathcal{M} : \{\mathbf{T}, \mathbf{F}\} \rightarrow \{\mathbf{W}, \mathbf{D}_{PSD}\}, \tag{2}$$

Where $\mathbf{T}$ are outfit template vertices and $\mathbf{F}$ is mesh faces, $\mathbf{W}$ and $\mathbf{D}_{PSD}$ are the blend weights and blend shapes matrices as defined above. Note that in deployment, a template outfit is processed by the network only once into its standard animated 3D model format. Once blend weights and blend matrices are obtained, the outfit is used as any other 3D animated model. This makes predictions automatically compatible with all graphics engines, and furthermore, due to the exhaustive optimization of rendering pipelines for such models, it is an extremely computationally efficient representation. This further extends its applicability to portable devices and low-computing environments. It represents an advantage against other related works that predict vertex locations directly with neural networks (and often through large, complex models). Such approaches require major engineering efforts to adapt to real applications. Furthermore, due to memory footprint and computational cost of neural networks, these solutions might be impossible to use in low-computing devices. Finally, we also show how this approach allows generalization to unseen template outfits without retraining, which greatly enhances scalability.

4. Methodology

Given PBS data for outfits on top of human bodies (SMPL) in different action sequences, we define samples $\mathcal{S} = \{X, Y\}$ as $X = \{\mathbf{T}, \mathbf{F}, \theta, \beta, g\}$ and $Y = \{\mathbf{V}_{PBS}\}$, where $\mathbf{T}$ is the template outfit vertices (canonical pose), $\mathbf{F}$ is outfit mesh faces, θ is body skeleton pose, β is body shape parameters, g is body gender and $\mathbf{V}_{PBS}$ is the outfit vertex locations in the simulated data. Our goal is to train $\mathcal{M}$ as defined in Eq. 2 such that $\mathbf{W}$ and $\mathbf{D}_{PSD}$ yield $\mathbf{V}_{PBS}$ after applying Eq. 1 (Note that for SMPL, skeleton is a function of shape β and gender).

4.1. PBS Data and Physical Consistency

The mapping from pose-space to outfit-space is a multi-valued function. Different simulators, initial conditions, action speeds, timesteps and integrators, among other factors, will generate different valid outfit vertex locations for the same body pose and shape and outfit. Training on PBS data

falsely assumes that this mapping is single-valued. Samples with similar X but significantly different Y will hinder network performance during training and most likely converge to average vertex location under a supervised loss. Moreover, a final user does not know the *ground truth* and therefore cannot perceive the accuracy of the model, but the user can assess the physical consistency of the predictions (collision-free and cloth consistency). Because of this, while resorting to PBS data for supervision is helpful for training networks, minimizing Euclidean error w.r.t. *ground truth* does not guarantee physical consistency, and therefore, the applicability of the predictions in real life is limited. Recent works [22, 27] propose post-processing to solve body penetrations. This partially defeats the purpose of using deep-learning and further compromises method compatibility and performance. We propose combining supervised training with unsupervised physically based training to alleviate the need of post-processing.

Physical consistency is a crucial part of proper outfit animation. While other approaches develop complex solutions to better overfit to their PBS data and translate training wrinkles to predictions, their lack of physical constraints is detrimental for their usability in real applications. Physical consistency is not only limited to collisions, but also to edge distortion and surface quality. Abnormally stretched or compressed edges (w.r.t. to its template lengths) will create texture distortions (UV map edges do not change in length, but mesh edges do). Approaches that represent garments as a subset of body vertices cannot enforce an edge constraint, as their template is the body itself (original template is lost after registration against the body). Our proposal addresses garment animation independently of edition/resizing, and therefore, it is possible to leverage the original template outfits to enforce the edge constraint during learning.

4.2. Architecture

The chosen architecture needs to be able to: a) handle unstructured meshes (no fixed vertex order or connectivity) and b) compute non-linear deformations w.r.t. the pose θ (as cloth behaviour is highly non-linear). To do so, we define the following components: $\Phi : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{N\times F}$, $\Omega : \mathbb{R}^{N\times F} \rightarrow \mathbb{R}^{N\times K}$, $\Psi : \mathbb{R}^{N\times F} \rightarrow \mathbb{R}^{P\times N\times 3}$ and $\chi : \mathbb{R}^{N\times F} \rightarrow \mathbb{R}^{P\times N\times 3}$. Component Φ computes per-vertex high-level F-dimensional descriptors with local and global information from template outfit mesh (with $F = 512$), Ω computes per-vertex blend weights from vertex descriptors, Ψ generates a blend shapes matrix supervisedly (note that it is equivalent to per-vertex *blend shapes* matrices as $\mathbf{d} \in \mathbb{R}^{P\times 3}$) and χ generates a blend shapes matrix unsupervisedly that will yield physical consistency. Note that we define P pose keys for blend shapes matrices, instead of the dimensionality of pose θ. We pass θ through an MLP to obtain a high-level embedding of pose as $\Theta \in \mathbb{R}^P$. The

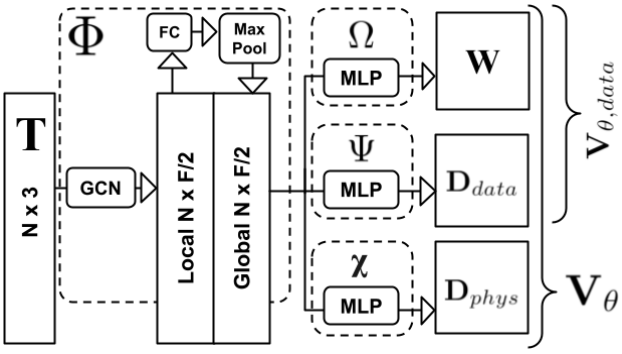

Figure 2: Model overview. The input of the model is a template outfit mesh (with no fixed topology, vertex order or connectivity). We apply graph convolutions to obtain vertex local descriptors. Then, local descriptors are processed by a fully-connected layer and aggregated through per-outfit max-pooling. This yields a global outfit descriptor that is concatenated with each vertex local descriptor. Final vertex descriptors are processed through different MLPs to obtain blend weights $\mathbf{W}$ and blend shapes matrices $\mathbf{D}_{data}$ and $\mathbf{D}_{phys}$. Blend shapes matrices are combined into $\mathbf{D}_{PSD}$, which is used as described in Eq. 1 to obtain final predictions. Pose keys for blend shapes matrix are obtained by passing θ through an MLP with 4 layers (not shown).

motivation for this is: a) controlling dimensionality P, and therefore, blend shapes matrix size and capacity and b) to allow modelling non-linearities from pose-space to vertex-space.

Fig. 2 shows an overview of the model. To learn Φ, we use 4 layers of graph convolutions applied to template mesh. This will yield a local descriptor, with no global information. Inspired by PointNet [25], we process each local descriptor through an additional fully-connected layer and aggregate all vertex descriptors through max-pooling (per outfit). We concatenate this global descriptor to each vertex local descriptor. Then, Ω, Ψ and χ are defined as MLPs, with 4 fully-connected layers each, applied to vertex descriptors (vertices are independent *samples*). The chosen architecture permits processing unstructured meshes with any vertex number, order and connectivity. This is a significant advantage against approaches that rely on the body model for garment representation [7, 22], since it requires an expensive registration for each sample that introduces error in the data. Then, Ψ and χ both compute blend shapes matrices: $\mathbf{D}_{data}$ to minimize supervision loss and $\mathbf{D}_{phys}$ for physical consistency. Despite being independent branches, on deployment, both matrices are combined to obtain the final PSD matrix $\mathbf{D}_{PSD} = \mathbf{D}_{data} + \mathbf{D}_{phys}$, thus keeping the aforementioned compatibility with graphics engines. Finally, the MLP used to obtain the high-level pose embedding Θ consists on 4 fully-connected layers. The output of

the model during training is $\mathbf{V}_{\theta,data}$ for $\mathbf{D}_{data}$ and $\mathbf{V}_\theta$ for $\mathbf{D}_{PSD}$.

4.3. Training

Our model combines both supervised and unsupervised training. The supervised part of the model corresponds to Φ, Ω and Ψ. The goal of this *submodel* is to minimize Euclidean error w.r.t. PBS data. Thus, for its training, we apply an standard L2 loss on predicted vertex locations:

$$\mathcal{L}_{data} = \sum \|\mathbf{V}_{\theta,data} - \mathbf{V}_{PBS}\|^2, \tag{3}$$

Then, the unsupervised part of the model corresponds only to χ. We define unsupervised losses to satisfy prior distributions based on physical constraints. First, to ensure cloth consistency of predictions, inspired by *mass-spring* model (most widely used PBS model for cloth), we define a cloth loss term as:

$$\mathcal{L}_{cloth} = \mathcal{L}_E + \lambda_B \mathcal{L}_B = \sum_{e \in E} \|e - e_{\mathbf{T}}\|^2 + \lambda_B \Delta(\mathbf{n})^2, \tag{4}$$

where $\mathcal{L}_E$ is the edge term and $\mathcal{L}_B$ is the bending term. Then, E is the set of edges of the given outfit mesh, e is the predicted edge length and $e_{\mathbf{T}}$ is the edge length on the template outfit $\mathbf{T}$. Then, $\Delta(\cdot)$ is the Laplace-Beltrami operator applied to vertex normals $\mathbf{n}$ of the predicted outfit and λ_B balances both losses. $\mathcal{L}_E$ enforces the output meshes to have the same edge lengths as the input template outfit, while $\mathcal{L}_B$ helps yielding locally smooth surfaces, as it penalizes differences on neighbouring vertex normals. To avoid excessive flattening, we choose $\lambda_B = 0.0005$. Then, to handle collisions against the body, we define a loss as:

$$\mathcal{L}_{collision} = \sum_{(i,j) \in A} min(\mathbf{d}_{j,i} \cdot \mathbf{n}_j - \epsilon, 0)^2, \tag{5}$$

where A is the set of correspondences (i, j) between predicted outfit and body through nearest neighbour, $\mathbf{d}_{j,i}$ is the vector going from the j-th vertex of the body to the i-th vertex of the outfit, $\mathbf{n}_j$ is the j-th vertex normal of the body and ϵ is a small positive threshold to increase robustness. This loss is a simplified formulation that assumes cloth is close to the skin, and penalizes outfit vertices placed inside the skin. In our experiments, we choose $\epsilon = 5$mm. Thus, the unsupervised loss is defined as:

$$\mathcal{L}_{phys} = \mathcal{L}_{cloth} + \lambda_{collision} \mathcal{L}_{collision} \tag{6}$$

where $\lambda_{collision}$ is the balancing weight for the collision term (around 2-10 in our experiments). Note how both terms $\mathcal{L}_{cloth}$ and $\mathcal{L}_{collision}$ are defined as priors (based only on X, not on Y). We define an additional loss term as an L2 regularization on deformations due to χ with a balancing weight $\lambda = 1e - 2$. This leads χ to use deformations as small as possible to solve physical constraints. While the whole model is differentiable and could be trained end-to-end, we back-propagate $\mathcal{L}_{phys}$ only through χ. The motivation for this is:

- **Independent Tasks.** We empirically observed how supervised and unsupervised terms fight each other, compromising one or both tasks. Thus, by training different parts of the model independently, we do not need to find a balance between low Euclidean error and physical consistency. This allows the supervised submodel to learn the main deformations leveraging PBS data and the unsupervised branch to enforce physical consistency without their gradients hindering each other.
- **Unsupervised Training.** Since $\mathcal{L}_{phys}$ does not rely on Y, it is possible to train χ with new samples where θ is replaced in X by any other sample pose. This increases the amount of available data to train, enhancing generalization of physical consistency.

In practice, it is not helpful to train χ until the supervised training has converged.

5. Experiments

From the public datasets on garments, only CLOTH3D [7] contains enough outfit variability to implement this approach and achieve proper generalization. It contains $\sim$ 7.5k sequences, each with a different template outfit in rest pose plus up to 300 frames. The outfits are simulated on top of an animated 3D human (SMPL), each with a different body shape. Likewise, we use SMPL skeleton in Eq. 1, so it drives the motion of the outfit, and its body mesh in Eq. 5. For the ablation study, we subsample 50k training *frames* and 5k test *frames* from CLOTH3D in a stratified manner w.r.t. sequences without outfit overlapping between both sets. Each model is trained for 10 epochs. We additionally present proof-of-concept computer vision applications as well as a performance analysis in the supplementary material.

5.1. Ablation study

We first evaluate the supervised part of the model (Φ, Ω and Ψ) by using the average vertex Euclidean error per outfit. In Tab. 1 we show the results to justify the design of the network. First, we propose a baseline model. In this baseline, global descriptor is not computed and Ψ predicts vertex deformations instead of blend shapes matrices by concatenating pose to vertex descriptors. The following models are modifications of the baseline (predict deformations). The second row shows the result obtained by using global descriptors. It improves the accuracy of the predictions. The third row corresponds to a model with a lower descriptor dimensionality ($F = 128$), and we observe an slight increase in error. In the next experiment, we implement Ω and Ψ as graph convolutions instead of fully-connected

Figure 3: Qualitative results obtained by enforcing physical consistency. For each sample we show the results of each experiment in Tab. 3 in the same order from left to right.

	Euclidean error (mm)
Baseline	29.98
+Global	28.04
+GlobalLite	28.59
+Global+GCN	28.76
+Global with MLP	28.43
DeePSD	**25.13**
-without pose embedding	30.93

Table 1: Architecture ablation study. First, as a baseline, we train Ω and Ψ to predict vertex deformations instead of blend shapes matrices. Subsequent rows are baselines extensions (deformation prediction) with a global descriptor. *DeePSD* row corresponds to the architecture shown in Fig. 2. As it can be seen, predicting blend shapes matrices is the best performing approach.

	Euclidean error (mm)
DeePSD	25.13
+ SMPL shape/gender	25.15
+ Fabric	24.76
+ Tightness + Fabric	**24.66**
+ SMPL + Tightness + Fabric	25.01

Table 2: Conditioning to metadata available in CLOTH3D [7] for each sample. We concatenate metadata to each vertex descriptor: SMPL shape and gender, per-garment fabric and per-outfit tightness. As shown, body metadata hinders performance, while outfit metadata enhances it.

layers. This worsens results at the cost of extra computational cost, thus we discard the use of graph convolutions after Φ. Note that this behaviour is expected, as global descriptor is broadcasted through vertices, and therefore, convolutions perform redundant information passes that hinder the learning. The next row corresponds to a model where global descriptor is obtained by replacing the single fully-connected layer in Φ with a MLP. Performance does not improve. *DeePSD* row corresponds to the architecture shown

	Error	Edge	Bend	Collision
No phys.	24.66	1.27	0.031	11.59%
Phys.	33.75	1.13	0.029	1.29%
+poses	34.45	1.12	0.029	1.02%

Table 3: Unsupervised training. We measure cloth quality with average edge elongation/compression and bending angle between neighbouring vertex normals. For body collision, we show the ratio of vertices placed within the body.

	Euclidean error (mm)
Tshirt	25.77
Top	17.33
Trousers	14.50
Jumpsuit	17.23
Skirt	41.15
Dress	35.94
Total	23.95

Table 4: Final quantitative results per garment. Note how *tighter* garment types have a significantly lower error than others.

in Fig. 2. As one can observe, predicting blend shapes matrices instead of vertex deformations not only increases model compatibility with graphics engines, but it also improves performance. The final row corresponds to the same architecture as DeePSD, but using pose θ as pose keys instead of a high-level pose embedding. We see that predictions are less accurate, thus pose embedding Θ is beneficial.

We consider the effect of including additional metadata present in CLOTH3D. That is, SMPL body shape and gender, garment-wise fabric labels and outfit-wise tightness values. We combine these metadata by concatenating them to each vertex descriptor. Tab. 2 shows the quantitative results. The first row corresponds to the best model of Tab. 1. Each next row is named after the metadata used. As it can be observed, outfit metadata reduces Euclidean error while body metadata appears to be detrimental.

To evaluate the unsupervised model, we design suitable metrics for assessing cloth quality and physical constraints:

- **Edge Length.** Length difference between predicted and rest outfit edges, expressed in millimeters.
- **Bend Angle.** Cosine distance for pairs of neighbouring vertex normals.
- **Collision.** Ratio of collided vertices.

Edge metric summarizes cloth integrity. Cloth needs to compress or stretch to fit its environment in real life and PBS, thus, a zero-valued edge error might be impossible (even undesirable). Nonetheless, an abnormally high value suggests distorted predictions. Similarly, bend angle cannot be zero, otherwise we have a completely flat surface. Again, high values for this metric show poor cloth quality. Finally, for collisions, a zero-valued metric means physically consistent predictions. In practice, the training data contains invalid combinations of pose and shape (bodies with self-collisions), and therefore, a 0% of collided vertices is impossible. Tab. 3 shows the results for the ablation study of the physical consistency. First, we evaluate the predictions obtained with supervised loss only (best model of Tab. 2). Second row shows the results obtained with χ trained without pose augmentation. The third row shows the results after training each sample with randomly chosen poses. We can observe that while Euclidean error increases, physical related metrics improve, specially collision. The model is learning to predict outfits farther from *ground truth* PBS data, but with higher physical consistency. As explained in Sec. 4.1, physical consistency cannot be summarized in one or few quantitative metrics. Results must be evaluated qualitatively. Fig. 3 shows a qualitative comparison of these experiments. As it can be seen, without physical constraints, although predictions have lower error by a large margin, qualitatively they are much worse. Also, we see that training unsupervisedly with randomly sampled poses further improves generalization.

We report final supervised results after fine-tuning with all data on Tab. 4. We decompose the error per garment. Note that *T-shirt* includes open shirts as well. We observe a worse performance for skirts and dresses. We also find a high error on *T-shirt*, likely due to open shirts. This is an expected behaviour, since modelling garments statically through skinning assumes the cloth will follow the body motion. Loose garments show a much more complex dynamics, and thus, static approaches will fail to model such garments. Fig. 1 shows qualitative results. We can see how the model can generalize to unseen complex outfits without retraining. Additionally, while cloth-to-cloth interaction is not explicitly addressed, the model is able to deal with multiple layers of cloth. It shows it can also handle complex geometric details (chest flower). As stated, it maintains cloth consistency, thus no artifacts appear on texturing. Finally, due to the unsupervised blend weights learning, skirts are robust against skinning artifacts due to leg motion (see supplementary material for more details on blend weights).

	Euclidean error (mm)
CLOTH3D [7]	29.0
DeePSD	**23.78**

Table 5: Comparison against CLOTH3D baseline. As CLOTH3D [7], we report the error garment-wise.

Figure 4: Qualitative comparison against CLOTH3D [7] baseline. Upper row: CLOTH3D. Lower row: DeePSD.

5.2. Comparison with related works

CLOTH3D. We compare DeePSD with CLOTH3D baseline quantitatively in Tab. 5 and qualitatively in Fig. 4. As it can be seen, our method outperforms CLOTH3D baseline. On the one hand, CLOTH3D baseline shows noisy boundaries and even broken suspenders. Furthermore, we observe the body geometry is present in the CLOTH3D reconstructed garment due to the use of SMPL body for garment representation. On the other hand, since DeePSD uses the original templates, boundaries are smooth and there is no bias to body geometry. Additionally, in spite of not dealing directly with cloth-to-cloth collisions, it appears that DeePSD is more robust in this aspect.

TailorNet. A fair quantitative comparison against the work of [22] is not possible. On one hand, TailorNet original simulations are not public, only the registered version against SMPL body. This means: a) original templates are lost and recovering them for each shape-style pair is unfeasible and b) their dataset has a fixed vertex order and connectivity (SMPL body). Since our main contribution is the generalization to unstructured meshes, comparing our methodology using a dataset with fixed vertex order against a methodology designed specifically for these data cannot be done fairly. On the the other hand, TailorNet is an ensemble of around 20 MLPs per each garment and gender which makes adapting it to CLOTH3D unfeasible, due to a much higher garment style variability. Thus, in Fig. 5, we compare TailorNet (left) and DeePSD (right) qualitatively. For fairness, since our approach uses no post-processing, we remove TailorNet post-processing.

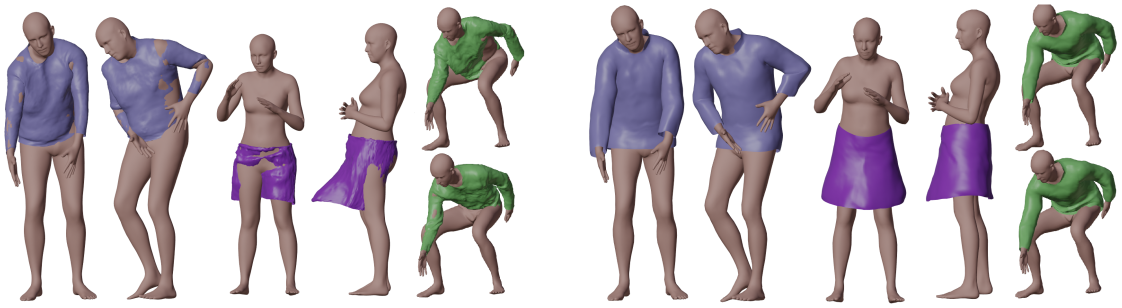

Figure 5: Comparison with TailorNet. Left: TailorNet. Right: DeePSD. TailorNet heavily relies in post-processing for valid predictions and generates noisy surfaces. The third sample (green T-shirt) shows two consecutive frames, note how TailorNet cannot guarantee temporal consistency.

We gather similar garments and body shapes in TailorNet data and CLOTH3D and compute the same sequences using both models. As it can be seen, TailorNet is highly dependant on its post-processing due to a high amount of collided vertices. For the green T-shirt, samples correspond to consecutive frames. TailorNet cannot keep temporal consistency. DeePSD does not suffer from such effect. Similar to CLOTH3D baseline, we observe how body geometry is present on TailorNet predictions (leftmost sample chest) due to the use of SMPL to represent garments.

TailorNet succeeds in generating wrinkles in their predictions by overfitting an ensemble of MLPs per each garment type and gender. As stated by its authors: *"Our key simplifying assumption is that two garments on two different people will deform similarly..."*. Nonetheless, this has drawbacks. On one hand, as we have seen, it strongly compromises physical consistency, and thus, relies on post-processing. This increases sample generation times by 150-300ms. Note that applying a post-processing eliminates differentiability. Another drawback is the complexity of their model. Their ensemble of MLPs takes around 2GB per garment and gender. All of this hurts its applicability, compatibility and performance (and then, portability). On the contrary, DeePSD is a single small-sized model (4.4MB) that allows animating any outfit (not only individual garments as body homotopies) without retraining. Predictions are generated as highly computationally efficient models (blend weights and blend shapes) compatible with any graphics engine. We obtain running times of 3-6ms for individual samples and around 0.1ms for batched samples (depending on vertex count). Furthermore, through physically based unsupervised learning, we alleviate the need of post-processing, thus maintaining differentiability and the aforementioned computational performance.

6. Conclusions and Future Work

We presented a novel approach for garment animation. Breaking the trend of previous approaches that try to predict vertex deformations through deep learning, we proposed learning a mapping from outfit space to animated 3D model space. We showed how this allows generalization to unseen outfits as well as compatibility with graphics engines. We observed how recent works need to leverage the body model for garment representation to allow edition/resizing along with animation, leading to overly complex models with scalability, compatibility and applicability issues. We addressed these issues by identifying garment animation as an independent task. We prioritized physical consistency in our predictions, relieving the need of post-processing. In summary, we developed an efficient approach applicable in real-scenarios as it is, even portable devices, that allows a more intuitive workflow for CGI artists that does not require expert knowledge in deep learning.

We observed limitations in our approach. First, loose garments, such as skirts and dresses, cannot be properly modelled with static approaches. To this end, we set as future work adapting our methodology to work with the temporal dimension. To keep its compatibility, pose keys should be computed with a temporal neural network while the training enforces dynamic learning (whether it is from data or unsupervisedly through physical consistency). We also observed how recent works grow on complexity to model fine geometric details (wrinkles). We believe the best approach to deal with garment wrinkles is through normal map generation because: a) it allows using lower vertex counts without compromising details, b) it is directly compatible with all graphics engines and c) it is more robust to collisions, since graphics engines compute face visibility on base geometry. Current works on this domain appear to be promising [14, 34]. We set this as future work.

Acknowledgements. This work has been partially supported by the Spanish project PID2019-105093GB-I00 (MINECO/FEDER, UE) and CERCA Programme/Generalitat de Catalunya.) This work is partially supported by ICREA under the ICREA Academia programme and Amazon Research Awards.

References

[1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8387–8397, 2018.

[2] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2293–2303, 2019.

[3] Brett Allen, Brian Curless, and Zoran Popović. Articulated body deformation from range scan data. *ACM Transactions on Graphics (TOG)*, 21(3):612–619, 2002.

[4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.

[5] Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D Kulkarni, and Joshua B Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1511–1519, 2017.

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

[7] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: Clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020.

[8] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5420–5430, 2019.

[9] Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. Drape: Dressing any person. *ACM Transactions on Graphics (TOG)*, 31(4):1–10, 2012.

[10] Xiaoguang Han, Chang Gao, and Yizhou Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on graphics (TOG)*, 36(4):1–12, 2017.

[11] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. *arXiv preprint arXiv:2004.00214*, 2020.

[12] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Geometric skinning with approximate dual quaternion blending. *ACM Transactions on Graphics (TOG)*, 27(4):1–23, 2008.

[13] Ladislav Kavan and Jiří Žára. Spherical blend skinning: a real-time deformation of articulated models. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 9–16, 2005.

[14] 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.

[15] Binh Huy Le and Zhigang Deng. Smooth skinning decomposition with rigid bones. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[16] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, 2000.

[17] Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Transactions on Graphics (TOG)*, 36(3):1–16, 2017.

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[19] Meysam Madadi, Hugo Bertiche, and Sergio Escalera. Smplr: Deep learning based smpl reverse for 3d human pose and shape recovery. *Pattern Recognition*, page 107472, 2020.

[20] Nadia Magnenat-thalmann, Richard Laperrire, Daniel Thalmann, and Université De Montréal. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface '88*, pages 26–33, 1988.

[21] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*, pages 484–494. IEEE, 2018.

[22] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7365–7375, 2020.

[23] Xavier Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation'97*, pages 177–189. Springer, 1997.

[24] 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.

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[26] Elad Richardson, Matan Sela, and Ron Kimmel. 3d face reconstruction by learning from synthetic data. In *2016 fourth international conference on 3D vision (3DV)*, pages 460–469. IEEE, 2016.

[27] Igor Santesteban, Miguel A Otaduy, and Dan Casas. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, volume 38, pages 355–366. Wiley Online Library, 2019.

[28] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep

learning for 3d object classification. In *Advances in neural information processing systems*, pages 656–664, 2012.

[29] Min Tang, Ruofeng Tong, Rahul Narain, Chang Meng, and Dinesh Manocha. A gpu-based streaming algorithm for high-resolution cloth simulation. In *Computer Graphics Forum*, volume 32, pages 21–30. Wiley Online Library, 2013.

[30] Tzvetomir Vassilev, Bernhard Spanlang, and Yiorgos Chrysanthou. Fast cloth animation on walking avatars. In *Computer Graphics Forum*, volume 20, pages 260–267. Wiley Online Library, 2001.

[31] Robert Y Wang, Kari Pulli, and Jovan Popović. Real-time enveloping with rotational regression. In *ACM SIGGRAPH 2007 papers*, pages 73–es. 2007.

[32] Xiaohuan Corina Wang and Cary Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 129–138, 2002.

[33] Cyril Zeller. Cloth simulation on the gpu. In *ACM SIGGRAPH 2005 Sketches*, pages 39–es. 2005.

[34] Meng Zhang, Tuanfeng Wang, Duygu Ceylan, and Niloy J Mitra. Deep detail enhancement for any garment. *arXiv e-prints*, pages arXiv–2008, 2020.