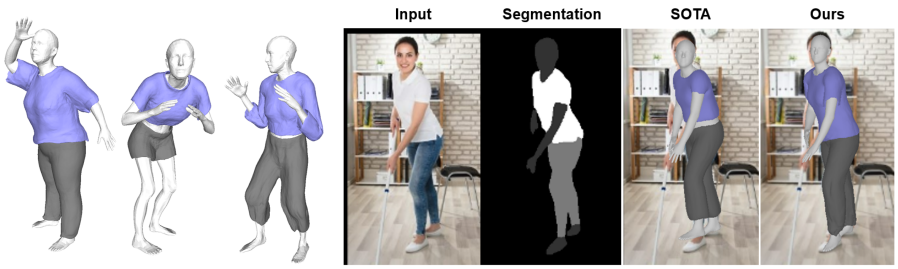
DIG: Draping Implicit Garment over the Human Body*

Ren Li[1][0000−0003−2998−7104], Benoît Guillard[1][0000−0002−8747−6153], Edoardo Remelli[2][0000−0002−8506−9191], and Pascal Fua[1][0000−0002−5477−1017]

[1] CVLab, EPFL, Switzerland
{ren.li, benoit.guillard, pascal.fua}@epfl.ch
[2] Meta Reality Labs Research, Zurich, Switzerland
edoremelli@fb.com



(a) (b)

Fig. 1. We introduce a pipeline for (a) generating and draping garments with various topology plausibly and (b) recovering garments from image observations (e.g. segmentation masks). Unlike prior works, our method allows for joint optimization of garment and body meshes, resulting in more faithful reconstruction.

Abstract. Existing data-driven methods for draping garments over human bodies, despite being effective, cannot handle garments of arbitrary topology and are typically not end-to-end differentiable. To address these limitations, we propose an end-to-end differentiable pipeline that represents garments using implicit surfaces and learns a skinning field conditioned on shape and pose parameters of an articulated body model. To limit body-garment interpenetrations and artifacts, we propose an interpenetration-aware pre-processing strategy of training data and a novel training loss that penalizes self-intersections while draping garments. We demonstrate that our method yields more accurate results for garment reconstruction and deformation with respect to state of the art methods. Furthermore, we show that our method, thanks to its end-to-end differentiability, allows to recover body and garments parameters jointly from image observations, something that previous work could not do. Our code is available at https://github.com/liren2515/DIG.

* This work was supported in part by the Swiss National Science Foundation.

1 Introduction

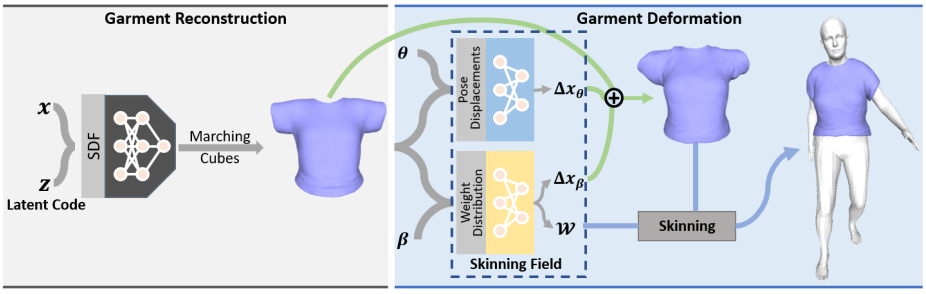


Fig. 2. The pipeline of our approach. The garment in the canonical space is first reconstructed from SDF. Given the shape β and pose θ of the target body, we add the shape and pose displacements (Δx_β and Δx_θ) to the reconstructed garment and drape it to the target body by the skinning function.

Modeling clothed humans has applications in industries such as fashion design, moviemaking, and video gaming. Many professional tools that rely on Physics-Based Simulation (PBS) [24,35,25,10] can be used to model cloth deformations realistically. However, they are computationally expensive, which precludes real-time use. Some of these can operate in near real-time using an incremental approach in motion sequences. However, these methods remain too slow for static cloth draping over a body in an arbitrary pose.

In recent years, there has therefore been considerable interest in using data-driven techniques to overcome these difficulties. They fall into two main categories. First there are those that use a single model to jointly represent the person and their clothes [21,32,7,37,6]. They produce visually appealing results but, because the body and garment are bound together, they do not make it easy to mix and match different bodies and clothing articles. Second, there are methods that represent the body and clothes separately. For example, in [14,27,33,36,2], deep learning is used to define skinning functions that can be used to deform the garments according to body motion. In [9], the explicit representation of clothes is replaced by an implicit one that relies on an inflated SDF surrounding the garment surface. It makes it possible to represent garments with many different topologies using a single model. To this end, it relies on the fact that garments follow the underlying body pose predictably. Hence, for each garment vertex, it uses the blending weights of the closest body vertex in the SMPL model [20]. Unfortunately, this step involves a search, which makes it both computationally expensive and non-differentiable.

In this paper, we propose the novel data-driven approach to skinning depicted by Fig. 2. As in [9], we represent the garments in terms of an inflated SDF but, instead of using the SMPL skinning model, we learn a garment-specific one. This

makes our approach both more expressive and fully-differentiable. To address the interpenetration issues caused by SDF inflation, we devised an interpenetration-aware data preprocessing for our training data. And to properly regularize the learned skinning field and to prevent self-intersections, we introduce a new loss term whose minimization prevents the creation of garment artifacts when the body deforms.

As a result, our method yields state-of-the-art results for both garment reconstruction and deformation. Its full differentiability makes it possible to fit both body and garments to partial observations. In other words, our pipeline can be used to simultaneously optimize the body and garment meshes, whereas earlier work [9] can only be used to optimize the garment.

2 Related Work

Most garment deformation approaches are either physics-based or data-driven. The physics-based algorithms [1,23,22,19,18] yield highly-realistic deformations but tend to be computationally demanding. The data-driven approaches are much less expensive at inference-time, sometimes at the cost of realism. Here we focus on those that are designed to drape a garment on a posed body.

Templates. In [27,33,36,5,15,4,34], individual garments are represented by separate triangulated 3D meshes. The topology of each one is fixed and a specific deformation function has to be learned. As a result, given the raw scan of a new garment with a different geometry from those already modeled—for example, a skirt as opposed to pants and shorts—expert knowledge is required to create the new template. Furthermore, the deformation model being garment-dependent makes these approaches impractical on large arrays of garments and, hence, ill-suited to real-world applications.

Point-Clouds. In [14] and DeePSD [2], the meshes are replaced by clouds of 3D points. The deformation is estimated for each point separately, making it possible to animate outfits of arbitrary topology and geometric complexity. However, the garment topology of these work is still non-differentiable because they rely on vertex connections from the template, which are fixed and pre-designed. This is addressed in [39] by using a point-cloud template with a fixed number of points densely sampled from the body mesh. This yields differentiability but the lack of point connections makes the reconstructed garments a group of unordered points instead of a surface with concrete physical properties.

Implicit Functions. Deep implicit functions [26,8] are good at representing surfaces whose topology can change while preserving differentiability [30,13]. SMPLicit [9] is the only work we know of that takes advantage of this to drape garments over bodies. As a result, the model can be fitted to real-world images. However, SMPLicit suffers several limitations. First, it does not handle the interpenetration between the body and garment. Second, it directly uses the blending weights of the closest vertices in the body model [20], which oversimplifies the dynamics and yields over-smoothed results. Finally, the optimization routines used to solve the fitting problem include approximations that produce inaccu-

racies and prevent the fitting result from being optimal. Our approach is in a similar spirit but overcomes these limitations.

3 Method

We start from an implicit surface model of the garment in a canonical pose, that is, draped over an average body in a T-pose, which we then deform to fit different body shapes and poses. This yields a fully differentiable pipeline that can be used for animation and modeling from images.

3.1 Garment Representation

Watertight surfaces of arbitrary topology can be represented very effectively by the zero crossings of a signed distance function (SDF)

$$f_\Theta(\mathbf{x}, \mathbf{z}) \longrightarrow \mathbb{R} \ , \tag{1}$$

where f is implemented by a neural network with weights Θ, $\mathbf{x} \in \mathbb{R}^3$ is a point in space, and $\mathbf{z}$ is a latent vector that parameterizes the surface shape [26]. However, clothes have openings in them and are not watertight. To nevertheless represent them in this manner, we can first compute unsigned distances to the surfaces, subtract a small ϵ value and treat the result as a signed distance function. This amounts to *inflating* the garments and representing them as watertight thin surfaces of thickness 2ϵ, as in [9,12]. Note that ϵ cannot be too small and must be larger than marching cube's step size, introducing an undesirable dependency between the field and how it is meshed.

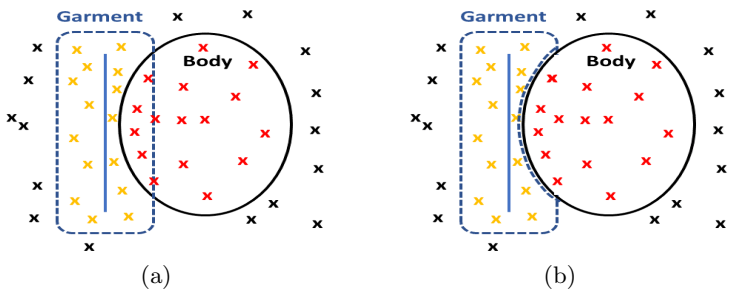
Given a database of garments fitted to a body in a T-pose shape and whose vertices coordinates have been normalized to be between -1 and 1, we use an auto-decoding approach to learning the weights Θ and the latent vectors $\mathbf{z}$ associated to specific garments. To this end, for each sample garment and its associated latent vector $\mathbf{z}$, we minimize a loss function

$$Loss = L_{SDF} + \lambda_{grad} L_{grad} + \lambda_{reg} \|\mathbf{z}\|^2 \ , \tag{2}$$

$$L_{SDF} = \sum_{\mathbf{x} \in X_v} \|f_\Theta(\mathbf{x}, \mathbf{z}) - s^{gt}(\mathbf{x})\| \ , \tag{3}$$

$$L_{grad} = \sum_{\mathbf{x} \in X_s} \|\nabla_x f_\Theta(\mathbf{x}, \mathbf{z}) - \mathbf{n}^{gt}(\mathbf{x})\|^2 + \sum_{\mathbf{x} \notin X_s} (\|\nabla_x f_\Theta(\mathbf{x}, \mathbf{z})\| - 1)^2 \ , \tag{4}$$

where s^{gt} and $\mathbf{n}^{gt}$ are ground-truth values of the signed distance function and normal, X_v and X_s represent points sampled in the $[-1, 1]^3$ volume and the garment surface respectively, and λ_{grad} and λ_{reg} are scalars that control the influence of the different terms. Minimizing L_{SDF} ensures that the SDF estimated by f_Θ is close to the ground-truth one in the whole volume while minimizing L_{grad} gives additional emphasis to it producing the right normals close to the surface and being a true SDF with unit gradients elsewhere, as in [11]. We present



(a) (b)

Fig. 3. The illustration of inflation processing for the garment surface (blue solid lines). (a) The inflation strategy of [12,9] will cause interpenetration between the inflated mesh (blue dashed line) and the body mesh, while (b) our proposed interpenetration-aware inflation will not.

an ablation study in the results section that shows that both are necessary to produce smooth and accurate surfaces.

One difficulty with this scheme arises from the fact that the garment is usually close to the underlying body mesh and inflating it by ϵ results in interpenetrations between garment and body, as shown in Fig. 3(a). Intersections between garments and the human body are problematic because they do not allow to employ the reconstructed meshes for downstream tasks such as e.g. physics simulations. Furthermore, in the experiment section, we show that learning a physically correct representation of garments where there are no interpenetration results in more accurate clothing deformations. To address this, we perform the interpenetration-aware pre-processing illustrated by Fig. 3(b) when sampling the surface points in the X_s set of Eq. 4. Given a garment mesh G, we sample a $256 \times 256 \times 256$ grid in $[-1, 1]^3$ to produce a set of points X and compute their signed distance to G. We then run Marching Cubes to recover the watertight mesh $M_{initial}$ as the dashed line of Fig. 3(a). For any vertex of $M_{initial}$ whose signed distance to the body is negative—meaning that it is inside it—we find the closest body vertex v_c and replace its position by $v_c + \mu \mathbf{n}_{v_c}$, where $\mathbf{n}_{v_c}$ is the surface normal at v_c and μ is a small positive value, which finally gives us the mesh M_{clean} without interpenetrations depicted by the blue dashed line of Fig. 3(b). In this example, X_s consists of points sampled from M_{clean} located on that dashed line. X_v comprises the points randomly sampled from $[-1, 1]^3$. Their position is not affected but their ground-truth signed distance is computed with respect to M_{clean}.

3.2 Modeling Garment Deformations

SMPL is a statistical parametric model that uses Linear Blend Skinning to deform a rigged body template $\mathbf{T} \in \mathbb{R}^{N_B \times 3}$ with N_B vertices. Given the parameters of shape β and pose θ, SMPL can generate the body mesh $M_B(\beta, \theta)$ by

$$M_B(\beta, \theta) = W(T_B(\beta, \theta), J(\beta), \theta, \mathcal{W}) \ , \tag{5}$$

$$T_B(\beta,\theta) = \mathbf{T} + B_s(\beta) + B_p(\theta)\ , \tag{6}$$

where W is the skinning function with weight $\mathcal{W} \in \mathbb{R}^{N_B \times 24}$ and joint locations $J(\beta) \in \mathbb{R}^{24\times 3}$. $B_s(\beta) \in \mathbb{R}^{N_B\times 3}$ and $B_p(\theta) \in \mathbb{R}^{N_B\times 3}$ are the shape and pose displacements. The SMPLicit algorithm [9] exploits the fact that the garment follows the pose of the underlying body in a predictable way by using for each garment vertex the blending weights of the closest body vertex. This step involves a search, which makes it both computationally expensive and non-differentiable.

To remedy this, we instead learn a specific blending model for the garment, which is different from that of the body. More specifically, we write

$$\begin{aligned} M_G(\mathbf{x},\beta,\theta) &= W(\mathbf{x}_{(\beta,\theta)}, J(\beta), \theta, \mathcal{W}(\mathbf{x}))\ , \\ \mathbf{x}_{(\beta,\theta)} &= \mathbf{x} + \Delta x_\beta(\mathbf{x}) + \Delta x_\theta(\mathbf{x})\ , \end{aligned} \tag{7}$$

where $W(\cdot)$ is the SMPL skinning function with learned skinning weights $\mathcal{W}(x) \in \mathbb{R}^{24}$, $\Delta x_\beta(\mathbf{x})$ and $\Delta x_\theta(\mathbf{x})$ are shape and pose displacements, and $\mathbf{x} \in \mathbb{R}^3$ denotes a generic 3D point instead of on a template. $\Delta x_\beta(\mathbf{x})$ models the shape offset conditioned on body shape β, while $\Delta x_\theta(\mathbf{x})$ represents a deformation field conditioned on body pose θ.

More specifically, $\mathcal{W}(\mathbf{x})$ and $\Delta x_\beta(\mathbf{x})$ are computed using the skinning weight $\mathcal{W}$ and shape displacement $B_s(\beta)$ from SMPL as base priors. They are extended to the whole 3D volume by writing

$$\mathcal{W}(\mathbf{x}) = w(\mathbf{x})\mathcal{W}, \quad \Delta x_\beta(\mathbf{x}) = w(\mathbf{x})B_s(\beta), \tag{8}$$

where $w(\mathbf{x}) \in \mathbb{R}^{N_B}$ are shared weights. Since $w(\cdot)$ is implemented by a neural network and $W(\cdot)$ is a differentiable function, M_G is fully differentiable, unlike the SMPLicit model [9]. The approach of [33] does something similar but in a more complex manner because it needs to learn separate models for blending weights and shape displacement, whereas we need only one. Furthermore, because $\mathbf{x}$ can be *any* 3D point, we can deform garments of arbitrary topology, instead of being restricted to a single garment template as in [27,15,33].

3.3 Training the Model

To train the network that implements the function w of Eq. 8, we use the same sampling strategy as in [33] to collect target $\bar{w}(x)$ values. For each $\mathbf{x} \in \mathbb{R}^3$, we sample N points $\mathcal{P} = \{\mathbf{p} : \mathbf{p} \sim \mathcal{N}(\mathbf{x}, d)\}$, where d is the distance from $\mathbf{x}$ to the body. We take $\bar{w}(\mathbf{x})$ to be

$$\bar{w}(\mathbf{x}) = \frac{1}{N}\sum_{\mathbf{p}\in\mathcal{P}} w_{bary}(\phi(\mathbf{p})), \tag{9}$$

where $\phi(\cdot)$ denotes the closest point on the body surface and $w_{bary}(\cdot)$ is a N_B-vector that uses the barycentric coordinate of the closest point as the weight for each body vertex. Since $\bar{w}(\mathbf{x})$ can be regarded as the weight distribution of body vertices, at training time, we introduce the loss

$$L_{KL} = \sum_x KL(w(\mathbf{x})||\bar{w}(\mathbf{x}))\ , \tag{10}$$

where KL is the KL-divergence. After the training of $w(x)$, we fix its parameter weights, plug it into our skinning model (Eq. 7), and then minimize the following loss for the training of Δx_θ

$$Loss = \lambda_{deform} L_{deform} + \lambda_{interp} L_{interp} + \lambda_{order} L_{order}. \tag{11}$$

where L_{interp} and L_{order} are regularization terms described below and λ_{deform}, λ_{interp}, and λ_{order} are scalar weights.

Dynamics. To capture detailed dynamics induced by pose changing, we define the deformation loss

$$L_{deform} = \sum_{\mathbf{x} \in X_s} |\bar{x}_d - \hat{x_d}(\mathbf{x})| + \sum_{\mathbf{x} \notin X_s} |\Delta x_\theta(\mathbf{x}) - \Delta x_\theta(\mathbf{x}_c)| \,, \tag{12}$$

where X_s denotes vertices of the ground-truth garment that forms an open surface, $\hat{x}_d$ and $\bar{x}_d$ are the point deformed according to Eq. 7 and the corresponding ground-truth position, respectively. $\mathbf{x}_c = \underset{\mathbf{x}' \in X_s}{\arg\min}\, d(\mathbf{x}', \mathbf{x})$ denotes the surface point closest to $\mathbf{x}$. As there are no correspondences in the training data for $\mathbf{x} \notin X_s$, the second term in Eq. 12 allows them be learned under the guidance of the closest surface points in the garment.

Interpenetrations. To prevent them, we utilize the SDF of the body mesh $M_B(\beta, \theta)$ to penalize the presence of deformed points inside the body. We write

$$L_{interp} = \sum_{\mathbf{x}} max(0, \epsilon_{SDF} - SDF_B(\hat{x_d}(\mathbf{x}))) \,, \tag{13}$$

where ϵ_{SDF} is a small value chosen to prevent $\hat{x_d}(\mathbf{x})$ from overlapping with the body surface.

Self-Intersections. Minimizing L_{deform} and L_{interp} usually suffices to deform open surfaces realistically. Unfortunately, when deforming the inflated watertight meshes we use, self-intersections can appear as shown on the left of Fig. 4(b). This can be understood as follows. Let us assume there are two points $\mathbf{x}_1$ and $\mathbf{x}_2$ on the inflated mesh whose closest surface point $\mathbf{x}_0$ is the same, as illustrated by Fig. 4(a). Let us further assume that $\mathbf{x}_2$ is initially farther from the body than $\mathbf{x}_1$. After deformation, nothing prevents $\mathbf{x}_1$ from ending up farther than $\mathbf{x}_2$ and yielding a self-intersection. To prevent this, we introduce the ordering loss

$$\begin{aligned} L_{order} = \sum_{(\mathbf{x}_1, \mathbf{x}_2) \in O} & max(0, SDF_B(\mathbf{x}_2 + \Delta x_\theta(\mathbf{x}_2)) - SDF_B(\mathbf{x}_1 + \Delta x_\theta(\mathbf{x}_1))) \,, \\ O = \{(x_1, x_2) | \psi(x_2) = \psi(x_1) & \text{ and } SDF_B(x_1) > SDF_B(x_2)\} \,, \end{aligned} \tag{14}$$

where $\psi(\cdot)$ denotes the closest garment vertex. Its minimization maintains the spatial relationship between points like $\mathbf{x}_1$ and $\mathbf{x}_2$ because it ensures that points, close to the body before deformation are still close after deformation.

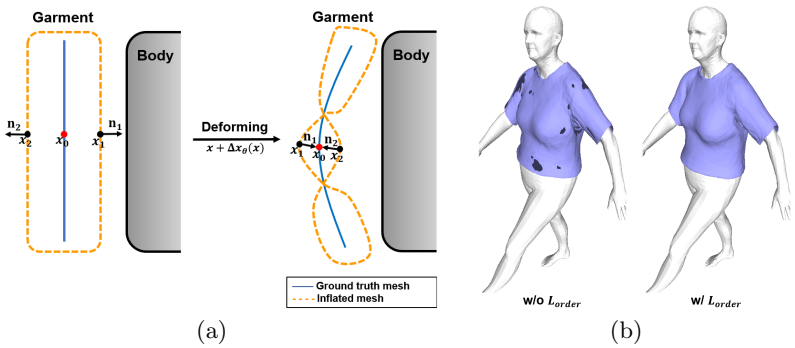


Fig. 4. (a) The illustration of how artifacts are produced by deformation. (b) Shirts deformed by the models trained w/ and w/o L_{order}. Without it, the inner face of the t-shirt can intersect the outer one. This is shown in dark blue.

3.4 Implementation Details

The SDF $f_\Theta(\mathbf{x}, \mathbf{z})$ of Eq. 1 is implemented by a 9-layer multilayer perceptron (MLP) with a skip connection from the input layer to the middle. We use Softplus as the activation function. The Θ weights and the latent code $z \in \mathbb{R}^{12}$ for each garment are optimized jointly during the training using a learning rate of 5e-4.

We use the architecture of [4] to implement the pose displacement network Δx_θ of Eq. 7. It comprises two MLP's with ReLU activation in-between. One encodes the pose θ to an embedding, and the other one predicts the blend matrices for the input point. The pose displacement is computed as the matrix product of the embedding and the blend matrices. The weight distribution $w(\cdot)$ of Eq. 8 is implemented by an MLP with an extra Softmax layer at the end to normalize the output. $N = 1000$ points are sampled to obtain the ground-truth $\bar{w}$ used to train w. We use the ADAM [16] optimizer with a learning rate of 1e-3 for the training of w and Δx_θ.

4 Experiments and Results

Our models can operate in several different ways. First, they can serve as generative models. By varying the latent code of our SDF f_Θ and using Marching Cubes, we can generate triangulated surfaces for garments of different topologies that can then be draped over bodies of changing shapes and poses using the deformation model M_G of Eq. 7, as shown in Fig. 1(a). Second, they can be used to recover both body *and* cloth shapes from images by minimizing

$$L(\beta, \theta, \mathbf{z}) = L_{\text{IoU}}(NeuR(M_G(\mathbf{G}, \beta, \theta), M_B(\beta, \theta)), \mathbf{S}) + L_{prior}(\theta) \ , \qquad (15)$$
$$\mathbf{G} = MC(f_\Theta(\mathbf{x}, \mathbf{z})) \ ,$$

where L_{IoU} is the IoU loss [17] that measures the difference between segmentation masks, $\mathbf{G}$ is the garment surface reconstructed by Marching Cubes $MC(\cdot)$,

$NeuR(\cdot)$ is a differentiable renderer, and $\mathbf{S}$ a semantic segmentation obtained using off-the-shelf algorithms. $M_B(\cdot)$ and $M_G(\cdot)$ are the skinning functions for garment and body as defined in Eq. 5 and 7. We also minimize the prior loss L_{prior} of VPoser [28] to ensure plausibility of the pose.

In theory $MC(\cdot)$ is not differentiable, but its gradient at vertex $\mathbf{x}$ can be approximated by $\frac{\partial \mathbf{x}}{\partial \mathbf{z}} = -\mathbf{n}\frac{\partial f(\mathbf{x},\mathbf{z})}{\partial \mathbf{z}}$, where $\mathbf{n} = \nabla f(\mathbf{x})$ is the normal [12]. In practice, this makes the minimization of Eq. 15 practical using standard gradient-based tools and we again rely on ADAM. Pytorch3D [29] serves as the differentiable renderer. In our experiments, we model shirts and trousers and use separate SDF and separate skinning models for each. Our pipeline can be used to jointly optimize the body mesh (β and θ) and the garment mesh ($\mathbf{z}$), while previous work [9] can only be used to optimize the garment.

In this section, we demonstrate both uses of our model. To this end, we first introduce the dataset and metrics used for our experiments. We then evaluate our method and compare its performance with baselines for garment reconstruction and deformation. Finally, we demonstrate the ability of our method to model people and their clothes from synthetic and real images.

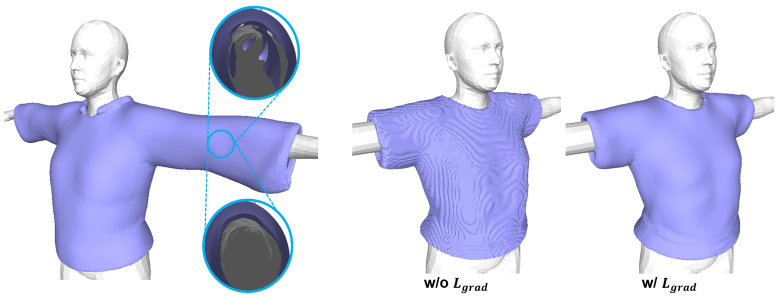
4.1 Dataset and Evaluation Metrics

We train our models on data from CLOTH3D [3]. It contains over 7k sequences of different garments draped on animated 3D human SMPL models. Each garment has a different template and a single motion sequence that is up to 300 frames long. We randomly select 100 shirts and 100 trousers, and transform them to a body with neutral-shape and T-pose by using displacement of the closest SMPL body vertex, which yields meshes in the canonical space. For each garment sequence, we use the first 90% frames as the training data and the rest as the test data (denoted as TEST EASY). We also randomly select 30 unseen sequences (denoted as TEST HARD) to test the generalization ability of our model. Chamfer Distance (CD), Euclidean Distance (ED), Normal Consistency (NC) and Interpenetration Ratio (IR) are reported as the evaluation metrics. NC is implemented as in [12]. IR is computed as the area ratio of garment faces inside the body to the overall garment faces.

4.2 Garment Reconstruction

The insets of Fig. 5(a) contrast our reconstruction results against those of SMPLicit [9]. The latter yields large interpenetrations while the former does not. Fig. 5(b) showcases the role of the L_{grad} term of Eq. 4 in producing smooth surfaces.

In Table 1, we report quantitative results for both the shirt and the trousers. We outperform SMPLicit (the first row - w/o *proc.*, w/o L_{grad}) in all three metrics. The margin in IR is over 18%, which showcases the ability of the interpenetration-aware processing of Section 3.1.



(a) (b)

Fig. 5. Reconstruction results. (a) The inside of the same garment reconstructed by SMPLicit (upper inset) and our method (bottom inset). The first features interpenetrations whereas the second does not. (b) Reconstructed garment by a model trained without and with L_{grad}. The latter is smoother and preserves details better.

Shirt	CD ($\times 10^{-4}$)	NC (%)	IR (%)
Ours w/o *proc.*, w/o L_{grad}	1.88	92.1	18.1
Ours w/o L_{grad}	1.58	90.3	**0.0**
Ours	**1.48**	**92.3**	**0.0**

Trousers	CD ($\times 10^{-4}$)	NC (%)	IR (%)
Ours w/o *proc.*, w/o L_{grad}	1.65	92.0	18.6
Ours w/o L_{grad}	**1.22**	91.8	**0.0**
Ours	1.34	**92.3**	**0.0**

Table 1. Comparative reconstruction results. *proc.* indicates our proposed interpenetration-aware pre-processing.

Shirt	ED (mm)	NC (%)	IR (%)
DeePSD	26.1	82.3	5.8
SMPLicit	35.9	84.0	13.3
Ours	**19.0**	**85.3**	**1.6**

Trousers	ED (mm)	NC (%)	IR (%)
DeePSD	17.5	85.4	1.5
SMPLicit	27.0	85.6	6.3
Ours	**14.8**	**86.7**	**0.2**

Table 2. Deforming unposed ground truth garments with DeePSD, SMPLicit and our method on TEST EASY.

Shirt	ED (mm)	NC (%)	IR (%)
DeePSD	95.6	72.6	46.4
SMPLicit	35.4	83.9	12.9
Ours	**26.5**	**85.1**	**3.0**

Trousers	ED (mm)	NC (%)	IR (%)
DeePSD	37.8	79.5	27.8
SMPLicit	31.9	84.9	8.8
Ours	**24.8**	**85.8**	**0.7**

Table 3. Deforming unposed ground truth garments with DeePSD, SMPLicit and our method on TEST HARD.

4.3 Garment Deformation

In this section, we compare our deformation results against those of SMPLicit [9] and DeePSD [2]. The input to DeePSD is the point cloud formed by the vertices of ground-truth mesh so that, like our algorithm, it can deform garments of

Shirt	CD ($\times 10^{-4}$)	NC (%)	IR (%)
SMPLicit	7.91	83.7	16.2
Ours - w/o *proc.*	3.86	84.4	1.6
Ours - w/ *proc.*	**3.78**	**84.7**	**1.5**

Trousers	CD ($\times 10^{-4}$)	NC (%)	IR (%)
SMPLicit	3.66	84.1	6.6
Ours - w/o *proc.*	2.71	85.3	**0.2**
Ours - w/ *proc.*	**2.67**	**85.4**	**0.2**

Table 4. Deforming SDF reconstructed garments with SMPLicit and our method on TEST EASY. w/o and w/ *proc.* means the mesh is reconstructed without and with interpenetration-aware processing respectively.

arbitrary topology by estimating the deformation for each point separately. To skin the garment, it learns functions to predict the blending weight and pose displacement. It also includes a self-consistency module to handle body-garment interpenetration. Hence, for a fair comparison, we retrain DeePSD using the same training data as before.

To test the deformation behavior of our model, we use the SMPL parameters β and θ provided by the test data as the input of our skinning model. As to the garment mesh to be deformed, we either use the ground-truth unposed mesh from the data, which is an open surface, or the corresponding watertight mesh reconstructed by our SDF model.

In Fig. 4(b), we presented a qualitative result that shows the importance of the ordering term of L_{order} in Eq. 14. We report quantitative results with the ground-truth mesh in Table 2 on TEST EASY. Our model performs substantially better than both baselines with the lowest ED and IR and the highest NC. For example, comparing to SMPLicit, the ED and IR of our model drop by more than 15mm and 10% for the deformation of shirt. In Table 3, we report similar results on TEST HARD, which is more challenging since it resembles less the training set, and we can draw the same conclusions. Since the learning of blending weights in DeePSD does not exploit the prior of the body model as us (Eq. 8), it suffers a huge performance deterioration in this case where its ED even goes up to 95.6mm and 37.8mm for the shirt and trousers respectively. Table 4 reports the results with SDF reconstructed mesh. Again, our method performs consistently better than SMPLicit in all metrics (row 2 vs row 4). It is also noteworthy that our interpenetration-aware pre-processing can help reduce deformation error and interpenetration ratio as indicated by the results of row 3 and 4. This demonstrates that learning a physically accurate model of garment interpenetrations results in more accurate clothing deformations.

In the qualitative results of Fig. 6, we can observe that SMPLicit cannot generate realistic dynamics and its results tend to be over-smoothed due to its simple skinning strategy. DeePSD can produce results that are better but too noisy. Besides, neither of them is able to address the body-garment interpenetration. Fig. 7 visualizes the level of interpenetrations happening different body region. We can notice that interpenetrations occur on almost everywhere in the body for SMPLicit. DeePSD shows less but still not as good as ours.

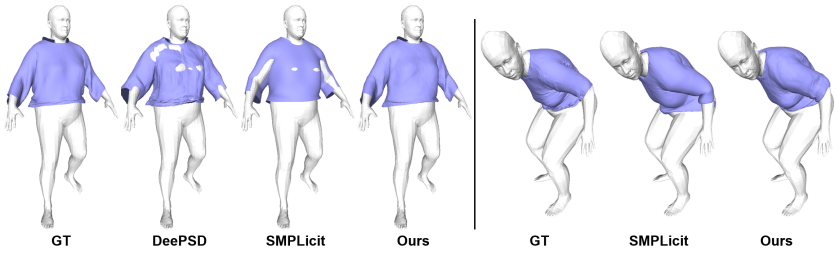
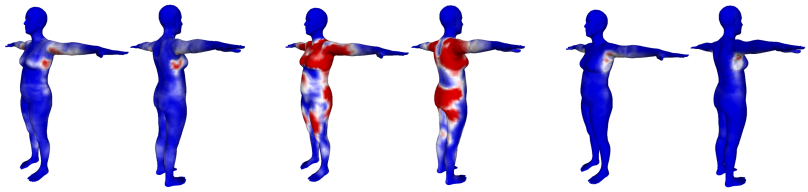


Fig. 6. The skinning results for the ground-truth shirt (left) and the SDF reconstructed shirt (right). Since the input of DeePSD should be the point cloud of the mesh template, we only evaluate it with the unposed ground-truth mesh. Compared to DeePSD and SMPLicit, our method can produce more realistic details and have less body-garment interpenetration.



(a) DeePSD (b) SMPLicit (c) Ours

Fig. 7. The visualization of the body region having interpenetrations (marked in red).

Shirt	CD ($\times 10^{-4}$)	NC (%)	IR (%)
SMPLicit-raw	17.77	82.1	41.5
SMPLicit	18.73	82.8	37.3
Ours	**4.69**	**87.3**	**3.9**

Trousers	CD ($\times 10^{-4}$)	NC (%)	IR (%)
SMPLicit-raw	4.22	81.2	35.7
SMPLicit	4.50	82.2	29.2
Ours	**2.23**	**89.2**	**0.7**

Table 5. The evaluation results of SMPLicit-raw (w/o smoothing), SMPLicit (w/ smoothing) and our method for garment fitting on the synthetic data.

4.4 From Images to Clothed People

Our model can be used to recover the body and garment shapes of clothed people from images by minimizing L of Eq. 15 with respect to β, θ, and $\mathbf{z}$. To demonstrate this, we use both synthetic and real images and compare our results to those of SMPLicit. Our optimizer directly uses the posed garment to compute the loss terms. In contrast, SMPLicit performs the optimization on the unposed garment. It first samples 3D points $\mathbf{p}$ in the canonical space. i.e. on the unposed body, and uses the weights of the closest body vertices to project these points into posed space and into 2D image space to determine if semantic label, 1 if

inside the garment, 0 otherwise. The loss

$$L(\mathbf{z}_G) = \begin{cases} |C(\mathbf{p}, \mathbf{z}_G) - \mathbf{d}_{max}|, & \text{if } s_\mathbf{p} = 0 \\ \min_i |C(\mathbf{p}^i, \mathbf{z}_G)|, & \text{if } s_\mathbf{p} = 1 \end{cases}, \tag{16}$$

is then minimized with respect to the latent code $\mathbf{z}_G$, where $\mathbf{d}_{max}$ is the maximum cut-off distance and $\min_i$ is used to consider only the point closest to the current garment surface estimate. This fairly complex processing chain tends to introduce inaccuracies.

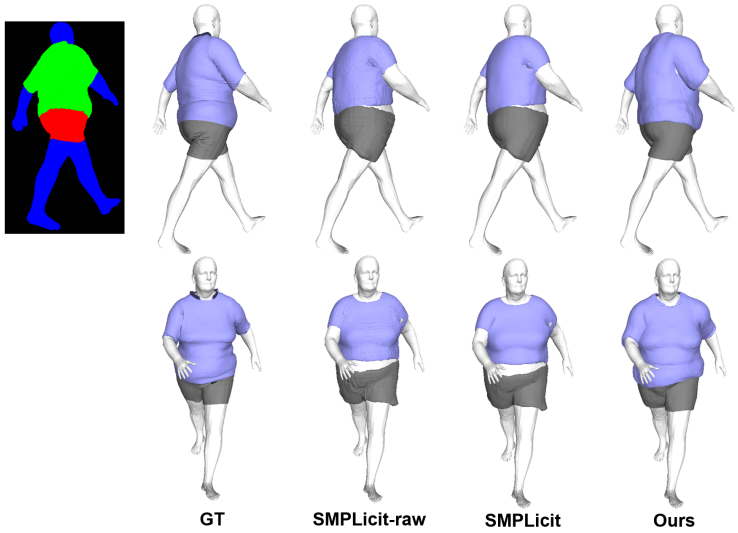


Fig. 8. Fitting results on a synthetic image. Left to right: the ground-truth segmentation and garment meshes, SMPLicit w/o smoothing (SMPLicit-raw), SMPLicit w/ smoothing (SMPLicit) and ours. Note that SMPLicit requires post-processing to remove artifacts, while our method does not.

Synthetic Images. We use the body and garment meshes from CLOTH3D as the synthetic data. Since the ground-truth SMPL parameters are available, we only optimize the latent code $\mathbf{z}$ for the garment and drop the pose prior term L_{prior} from Eq. 15. Image segmentation such as the one of Fig. 8 are obtained by using Pytorch3D to render meshes under specific camera configurations. Given the ground-truth β, θ and segmentation, we initialize $\mathbf{z}$ as the mean of learned codes and then minimize the loss. Fig 8 shows qualitative results in one specific case. The quantitative results reported in Table 5 confirm the greater accuracy and lesser propensity to produce interpenetrations of our approach.

Real Images. In real-world scenarios such as those depicted by Fig. 9, there are no ground-truth annotations but we can get the required information from single

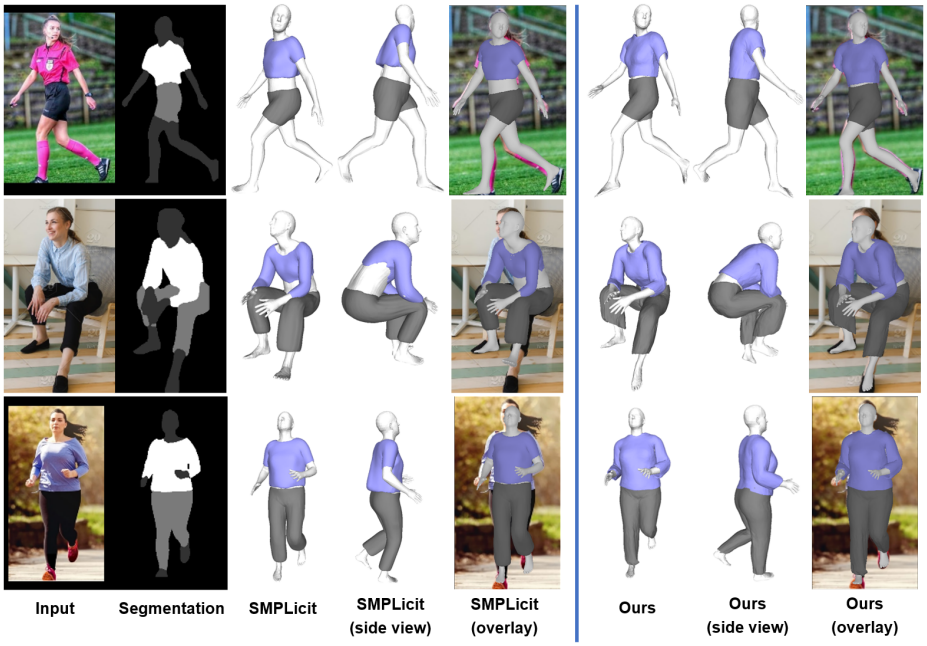


Fig. 9. Fitting results on images in-the-wild. Left to right: the input images and their segmentation, SMPLicit and ours. Note that SMPLicit recovers garments based on body meshes estimated from [31], while we can optimize the body and garment parameters jointly for more accurate results.

images from off-the-shelf algorithms. As in SMPLicit, we use [31] to estimate the SMPL parameters $\hat{\beta}$ and $\hat{\theta}$ and the algorithm of [38] to produce a segmentation. In SMPLicit, $\hat{\beta}$ and $\hat{\theta}$ are fixed and only the garment model is updated. In contrast, in our approach, $\hat{\beta}$, $\hat{\theta}$, and the latent vector $\mathbf{z}$ are all optimized. As can be seen in Fig. 9, this means that inaccuracies in the $\hat{\beta}$ and $\hat{\theta}$ initial values can be corrected, resulting in an overall better fit of both body and garments.

5 Conclusion

We have presented a fully differentiable approach to draping a garment on a body so that both body and garment parameters can be jointly optimized. At its heart is a skinning model that learns to prevent self-penetration. We have demonstrated its effectiveness both for animation purposes and to recover body and cloth shapes from real images. In future work, we will incorporate additional physics-based constraints to increase realism and to reduce the required amount of training data.

References

1. Baraff, D., Witkin, A.: Large Steps in Cloth Simulation. In: ACM SIGGRAPH. pp. 43–54 (1998)
2. Bertiche, H., Madadi, M., Tylson, E., Escalera, S.: DeePSD: Automatic Deep Skinning and Pose Space Deformation for 3D Garment Animation. In: International Conference on Computer Vision (2021)
3. Bertiche, H., Madadi, M., Escalera, S.: CLOTH3D: clothed 3d humans. In: European Conference on Computer Vision. pp. 344–359 (2020)
4. Bertiche, H., Madadi, M., Escalera, S.: PBNS: physically based neural simulation for unsupervised garment pose space deformation. ACM Transactions on Graphics **40**(6), 1–14 (2021)
5. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-Garment Net: Learning to Dress 3D People from Images. In: International Conference on Computer Vision (2019)
6. Chen, X., Jiang, T., Song, J., Yang, J., Black, M.J., Geiger, A., Hilliges, O.: gDNA: Towards Generative Detailed Neural Avatars. In: arXiv Preprint (2022)
7. Chen, X., Zheng, Y., Black, M.J., Hilliges, O., Geiger, A.: SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In: International Conference on Computer Vision. pp. 11594–11604 (2021)
8. Chibane, J., Mir, A., Pons-Moll, G.: Neural Unsigned Distance Fields for Implicit Function Learning. In: Advances in Neural Information Processing Systems (2020)
9. Corona, E., Pumarola, A., Alenya, G., Pons-Moll, G., Moreno-Noguer, F.: Smplicit: Topology-Aware Generative Model for Clothed People. In: Conference on Computer Vision and Pattern Recognition (2021)
10. Designer, M.: (2018), `https://www.marvelousdesigner.com`
11. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit Geometric Regularization for Learning Shapes. In: International Conference on Machine Learning (2020)
12. Guillard, B., Remelli, E., Lukoianov, A., Richter, S., Bagautdinov, T., Baque, P., Fua, P.: Deepmesh: Differentiable Iso-Surface Extraction. In: arXiv Preprint (2021)
13. Guillard, B., Stella, F., Fua, P.: MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks. In: arXiv Preprint (2021)
14. Gundogdu, E., Constantin, V., Parashar, S., Seifoddini, A., Dang, M., Salzmann, M., Fua, P.: Garnet++: Improving Fast and Accurate Static 3D Cloth Draping by Curvature Loss. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(1), 181–195 (2022)
15. Jiang, B., Zhang, J., Hong, Y., Luo, J., Liu, L., Bao, H.: Bcnet: Learning body and cloth shape from a single image. In: European Conference on Computer Vision. pp. 18–35 (2020)
16. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: International Conference on Learning Representations (2015)
17. Li, R., Zheng, M., Karanam, S., Chen, T., Wu, Z.: Everybody Is Unique: Towards Unbiased Human Mesh Recovery. In: British Machine Vision Conference (2021)
18. Li, Y., Habermann, M., Thomaszewski, B., Coros, S., Beeler, T., Theobalt, C.: Deep physics-aware inference of cloth deformation for monocular human performance capture. In: International Conference on 3D Vision. pp. 373–384 (2021)
19. Liang, J., Lin, M., Koltun, V.: Differentiable Cloth Simulation for Inverse Problems. In: Advances in Neural Information Processing Systems (2019)

20. Loper, M., Black, M.: Opendr: An Approximate Differentiable Renderer. In: European Conference on Computer Vision. pp. 154–169 (2014)
21. Ma, Q., Saito, S., Yang, J., Tang, S., Black, M.J.: SCALE: Modeling Clothed Humans with a Surface Codec of Articulated Local Elements. In: Conference on Computer Vision and Pattern Recognition (2021)
22. Narain, R., Pfaff, T., O'Brien, J.F.: Folding and crumpling adaptive sheets. ACM Transactions on Graphics **32**(4), 1–8 (2013)
23. Narain, R., Samii, A., O'brien, J.F.: Adaptive anisotropic remeshing for cloth simulation. ACM Transactions on Graphics **31**(6), 1–10 (2012)
24. Nvidia: Nvcloth (2018)
25. Nvidia: NVIDIA Flex (2018), `https://developer.nvidia.com/flex`
26. Park, J.J., Florence, P., Straub, J., Newcombe, R.A., Lovegrove, S.: DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In: Conference on Computer Vision and Pattern Recognition (2019)
27. Patel, C., Liao, Z., Pons-Moll, G.: Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In: Conference on Computer Vision and Pattern Recognition. pp. 7365–7375 (2020)
28. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In: Conference on Computer Vision and Pattern Recognition. pp. 10975–10985 (2019)
29. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: PyTorch3D. `https://github.com/facebookresearch/pytorch3d` (2020)
30. Remelli, E., Lukoianov, A., Richter, S., Guillard, B., Bagautdinov, T., Baque, P., Fua, P.: Meshsdf: Differentiable Iso-Surface Extraction. In: Advances in Neural Information Processing Systems (2020)
31. Rong, Y., Shiratori, T., Joo, H.: Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. In: arXiv Preprint (2020)
32. Saito, S., Yang, J., Ma, Q., Black, M.J.: SCANimate: Weakly supervised learning of skinned clothed avatar networks. In: Conference on Computer Vision and Pattern Recognition. pp. 2886–2897 (2021)
33. Santesteban, I., Thuerey, N., Otaduy, M.A., Casas, D.: Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. In: Conference on Computer Vision and Pattern Recognition (2021)
34. Santesteban, I., Otaduy, M.A., Casas, D.: SNUG: Self-Supervised Neural Dynamic Garments. In: arXiv Preprint (2022)
35. Software, O.F.D.: (2018), `https://optitex.com/`
36. Tiwari, G., Bhatnagar, B.L., Tung, T., Pons-Moll, G.: Sizer: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing. In: European Conference on Computer Vision (2020)
37. Tiwari, G., Sarafianos, N., Tung, T., Pons-Moll, G.: Neural-GIF: Neural generalized implicit functions for animating people in clothing. In: International Conference on Computer Vision. pp. 11708–11718 (2021)
38. Yang, L., Song, Q., Wang, Z., Hu, M., Liu, C., Xin, X., Jia, W., Xu, S.: Renovating parsing R-CNN for accurate multiple human parsing. In: European Conference on Computer Vision. pp. 421–437 (2020)
39. Zakharkin, I., Mazur, K., Grigorev, A., Lempitsky, V.: Point-based modeling of human clothing. In: International Conference on Computer Vision. pp. 14718–14727 (2021)