DrapeNet: Garment Generation and Self-Supervised Draping

Luca De Luigi*,2 Ren Li*,1 Benoît Guillard[1] Mathieu Salzmann[1] Pascal Fua[1]

[1]: CVLab, EPFL, {name.surname}@epfl.ch [2]: University of Bologna, luca.deluigi4@unibo.it

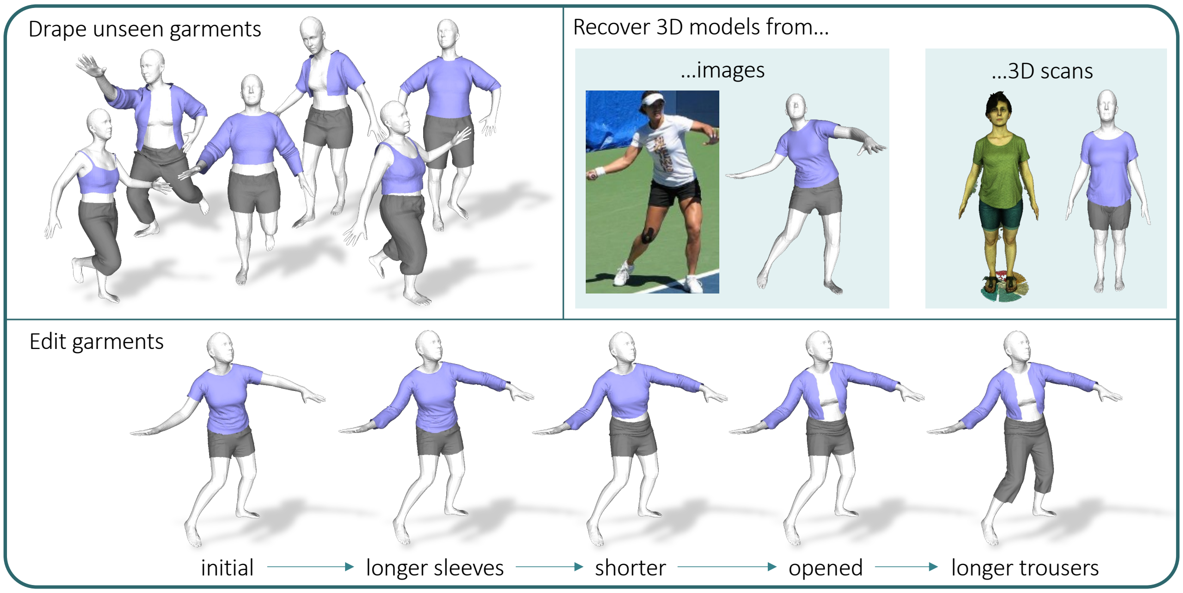

Figure 1. Our network can drape garments over bodies of different shapes in various poses. To minimize the required amount of supervision, our draping network is trained with physics-based self-supervision and generalizes to multiple garments by being conditioned on latent codes. These can be manipulated to edit specific features of the corresponding garments. Being fully differentiable, our pipeline makes it possible to recover 3D models of garments and bodies from observations such as images and 3D scans.

Abstract

Recent approaches to drape garments quickly over arbitrary human bodies leverage self-supervision to eliminate the need for large training sets. However, they are designed to train one network per clothing item, which severely limits their generalization abilities. In our work, we rely on self-supervision to train a single network to drape multiple garments. This is achieved by predicting a 3D deformation field conditioned on the latent codes of a generative network, which models garments as unsigned distance fields.

Our pipeline can generate and drape previously unseen garments of any topology, whose shape can be edited by manipulating their latent codes. Being fully differentiable, our formulation makes it possible to recover accurate 3D models of garments from partial observations – images or 3D scans – via gradient descent. Our code is publicly available at https://github.com/liren2515/DrapeNet.

1. Introduction

Draping digital garments over differently-shaped bodies in random poses has been extensively studied due to its many applications such as fashion design, moviemaking, video gaming, virtual try-on and, nowadays, virtual and augmented reality. Physics-based simulation (PBS) [3, 12, 23, 32, 33, 38, 39, 47–49, 51, 60] can produce outstanding results, but at a high computational cost.

Recent years have witnessed the emergence of deep neural networks aiming to achieve the quality of PBS draping while being much faster, easily differentiable, and offering new speed vs. accuracy tradeoffs [15, 16, 25, 36, 44, 46, 50, 53, 55]. These networks are often trained to produce garments that resemble ground-truth ones. While effective, this requires building training datasets, consisting of ground-truth meshes obtained either from computationally expensive simulations [31] or using complex 3D scanning setups [37]. Moreover, to generalize to unseen garments and poses, these supervised approaches require training databases encompassing a great variety of samples de-

[*] Equal contributions

picting many combinations of garments, bodies and poses.

The recent PBNS and SNUG approaches [5, 42] address this by casting the physical models adopted in PBS into constraints used for self-supervision of deep learning models. This makes it possible to train the network on a multitude of body shapes and poses without ground-truth draped garments. Instead, the predicted garments are constrained to obey physics-based rules. However, both PBNS and SNUG, require training a separate network for each garment. They rely on mesh templates for garment representation and feature one output per mesh vertex. Thus, they cannot handle meshes with different topologies, even for the same garment. This makes them very specialized and limits their applicability to large garment collections as a new network must be trained for each new clothing item.

In this work, we introduce `DrapeNet`, an approach that also relies on physics-based constraints to provide self-supervision but can handle generic garments by conditioning a *single* draping network with a latent code describing the garment to be draped. We achieve this by coupling the draping network with a garment *generative* network, composed of an encoder and a decoder. The encoder is trained to compress input garments into compact latent codes that are used as input condition for the draping network. The decoder, instead, reconstructs a 3D garment model from its latent code, thus allowing us to sample and edit new garments from the learned latent space.

Specifically, we model the output of the garment decoder as an unsigned distance function (UDF), which were demonstrated [14] to yield better accuracy and fewer interpenetrations than the inflated signed distance functions often used for this purpose [10, 19]. Moreover, UDFs can be triangulated in a differentiable way [14] to produce explicit surfaces that can easily be post-processed, making our pipeline fully differentiable. Hence, `DrapeNet` can not only drape garments over given body shapes but can also perform gradient-based optimization to fit garments, along with body shapes and poses, to partial observations of clothed people, such as images or 3D scans.

Our contributions are as follows:

- We introduce a *single* garment draping network conditioned on a latent code to handle generic garments from a large collection (e.g. *top* or *bottom* garments);
- By exploiting physics-based self-supervision, our pipeline only requires a few hundred garment meshes in a canonical pose for training;
- Our framework enables the fast draping of new garments with high fidelity, as well as the sampling and editing of new garments from the learned latent space;
- Being fully differentiable, our method can be used to recover accurate 3D models of clothed people from images and 3D scans.

2. Related Work

Implicit Neural Representations for 3D Surfaces. Implicit neural representations have emerged a few years ago as an effective tool to represent surfaces whose topology is not known a priori. They can be implemented using (clipped) *signed distance functions* (SDF) [35] or *occupancies* [28]. When an explicit representation is required, it can be obtained using Marching Cubes [18] and this can be done while preserving differentiability [1, 27, 41]. However, they can only represent watertight surfaces.

Thus, to represent open surfaces, such as clothes, it is possible to use inflated SDFs surrounding them. However, this entails a loss in accuracy and there has been a recent push to replace SDFs by *unsigned distance functions* (UDFs) [9, 52, 61]. One difficulty in so doing was that Marching Cubes was not designed with UDFs in mind, and obtaining explicit surfaces from these UDFs was therefore non-trivial. This has been addressed in [14] by modifying the Marching Cubes algorithm to operate with UDFs. We model garment with UDFs and use [14] to mesh them. Other works augment signed distance fields with covariant fields to encode open surface garments [8, 43].

Draping Garments over 3D Bodies. Two main classes of methods coexist, physics-based algorithms [3, 21, 22, 30, 31, 48] that produce high-quality drapings but at a high computational cost, and data-driven approaches that are faster but often at the cost of realism.

Among the latter, template-based approaches [5, 7, 17, 34, 36, 42, 45, 50] are dominant. Each garment is modeled by a specific triangulated mesh and a draping function is learned for each one. In other words, they do not generalize. There are however a number of exceptions. In [6, 15] the mesh is replaced by 3D point clouds that can represent generic garments. This enables deforming garments with arbitrary topology and geometric complexity, by estimating the deformation separately for each point. [59] goes further and allows differentiable changes in garment topology by sampling a fixed number of points from the body mesh. Unfortunately, this point cloud representation severely limits possible downstream applications.

In recent approaches [10, 19], a space of garments is learned with clothing items modeled as inflated SDFs and one single shared network to predict their deformations as a 3D displacement field. This makes deployment in real-world scenarios easier and allows the reconstruction of garments from images and 3D scans. However, the inflated SDF scheme reduces realism and precludes post-processing using standard physics-based simulators or other cloth-specific downstream applications. Furthermore, both models are fully supervised and require a dataset of draped garments whose collection is extremely time-consuming.

Alleviating the need for costly ground-truth draped garments is tackled in [5, 42], by introducing physics-based

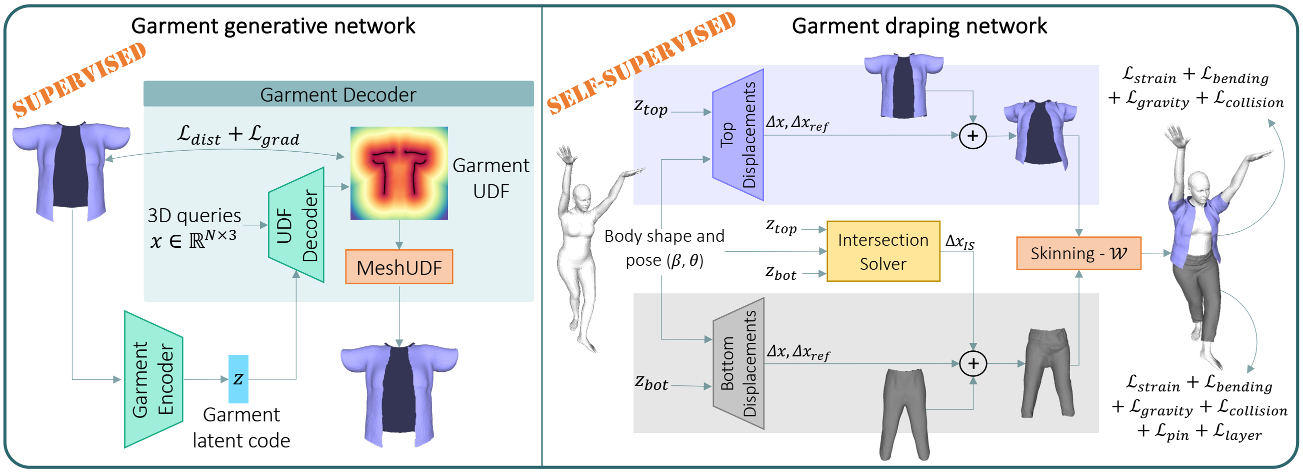
Garment generative network
SUPERVISED
Garment Decoder
$\mathcal{L}_{dist} + \mathcal{L}_{grad}$
Garment UDF
3D queries $x \in \mathbb{R}^{N\times3}$
UDF Decoder
MeshUDF
Garment Encoder
z
Garment latent code
Garment draping network
SELF-SUPERVISED
z_{top}
Top Displacements
$\Delta x, \Delta x_{ref}$
$\mathcal{L}_{strain} + \mathcal{L}_{bending} + \mathcal{L}_{gravity} + \mathcal{L}_{collision}$
Body shape and pose (β, θ)
z_{top}
z_{bot}
Intersection Solver
Δx_{IS}
Skinning - $\mathcal{W}$
z_{bot}
Bottom Displacements
$\Delta x, \Delta x_{ref}$
$\mathcal{L}_{strain} + \mathcal{L}_{bending} + \mathcal{L}_{gravity} + \mathcal{L}_{collision} + \mathcal{L}_{pin} + \mathcal{L}_{layer}$

Figure 2. **Overview of our framework. Left:** Garment generative network, trained to embed garments into compact latent codes and predict their unsigned distance field (UDF) from such vectors. UDFs are then meshed using [14]. **Right:** Garment draping network, conditioned on the latent codes of the generative network. It is trained in a self-supervised way to predict the displacements Δx and Δx_{ref} to be applied to the vertices of given garments, before skinning them according to body shape and pose (β, θ) with the predicted blending weights $\mathcal{W}$. It includes an Intersection Solver module to prevent intersection between top and bottom garments.

losses to train draping networks in a self-supervised manner. The approach of [42] relies on a mass spring model to enforce the physical consistency of static garments deformed by different body poses. The method of [5] also accounts for variable body shapes and dynamic effects; furthermore, it incorporates a more realistic and expressive material model. Both methods, however, require training one network per garment, a limitation we remove.

3. Method

We aim to realistically deform and drape generic garments over human bodies of various shapes and poses. To this end, we introduce the `DrapeNet` framework, presented in Fig. 2. It comprises a generative network shown on the left and a draping network shown on the right. Only the first is trained in a supervised manner, but using only static unposed garments meshes. This is key to avoiding having to run physics-based simulations to generate ground-truth data. Furthermore, we condition the draping network on latent vectors representing the input garments, which allows us to use the same network for very different garments, something that competing methods [5, 42] cannot do.

The generative network is a decoder trained using an encoder that turns a garment into a latent code $\mathbf{z}$ that can then be decoded to an Unsigned Distance Function (UDF), from which a triangulated mesh can be extracted in a differentiable manner [14]. The UDF representation allows us to accurately represent open surfaces and the many openings that garments typically feature. Since the top and bottom garments – shirts and trousers – have different patterns, we train one generative model for each. Both networks have the same architecture but different weights.

The resulting *garment generative network* is only trained to output garments in a canonical shape, pose, and size that fit a neutral SMPL [24] body. Draping the resulting garments to bodies in non-canonical poses is then entrusted to a *draping network*, again one for the top and one for the bottom. As in [5, 19, 42], this network predicts vertex displacements w.r.t. the neutral position. The deformed garment is then skinned onto the articulated body model. To enable generalization to different tops and bottoms, we condition the draping process on the garment latent codes of the generative network, shown as $\mathbf{z}_{\text{top}}$ and $\mathbf{z}_{\text{bot}}$ in Fig. 2.

We use a small database of static unposed garments loosely aligned with bodies in the canonical position to train the two garment generating networks. This being done, we exploit physics-based constraints to train in a fully self-supervised manner the top and bottom draping networks for realism, without interpenetrations with the body and between the garments themselves.

3.1. Garment Generative Network

To encode garments into latent codes that can then be decoded into UDFs, we rely on a point cloud encoder that embeds points sampled from the unposed garment surface into a compact vector. This lets us obtain latent codes for previously unseen garments in a single inference pass from points sampled from its surface. This can be done given any arbitrary surface triangulation. Hence, it gives us the flexibility to operate on any given garment mesh.

We use DGCNN [56] as the encoder. It first propagates the features of points within the same local region at multiple scales and then aggregates them into a single global embedding by max pooling. We pair it with a decoder that takes as input a latent vector, along with a point in 3D space, and returns its (unsigned) distance to the garment. The decoder is a multi-layer perceptron (MLP) that relies on Conditional Batch Normalization [54] for conditioning on the input latent vector.

We train the encoder and the decoder by encouraging them to jointly predict distances that are small near the training garments' surface and large elsewhere. Because the algorithm we use to compute triangulated meshes from the predicted distances [14] relies on the gradient vectors of the UDF field, we also want these gradients to be as accurate as possible [2, 61]. We therefore minimize the loss

$$L_{garm} = L_{dist} + \lambda_g L_{grad} \; , \tag{1}$$

where L_{dist} encodes our distance requirements, L_{grad} the gradient ones, and λ_g is a weight balancing their influence.

More formally, at training time and given a mini-batch comprising B garments, we sample a fixed number P of points from the surface of each one. For each resulting point cloud $\mathbf{p}_i$ ($1 \leq i \leq B$), we use the garment encoder E_G to compute the latent code

$$\mathbf{z}_i = E_G(\mathbf{p}_i) \tag{2}$$

and use it as input to the decoder D_G. It predicts an UDF field supervised with Eq. (1), whose terms we define below.

Distance Loss. Having experimented with many different formulations of this loss, we found the following one both simple and effective. Given N points $\{\mathbf{x}_{ij}\}_{j \leq N}$ sampled from the space surrounding the i-th garment, we pick a distance threshold δ, clip all the ground-truth distance values $\{y_{ij}\}$ to it, and linearly normalize the clipped values to the range $[0, 1]$. This yields normalized ground-truth values $\bar{y}_{ij} = \min(y_{ij}, \delta)/\delta$. Similarly, we pass the output of the final layer of D_G through a sigmoid function $\sigma(\cdot)$ to produce a prediction in the same range for point $\mathbf{x}_{ij}$

$$\widetilde{y}_{ij} = \sigma(D_G(\mathbf{x}_{ij}, \mathbf{z}_i)) \; . \tag{3}$$

Finally, we take the loss to be

$$\mathcal{L}_{dist} = \text{BCE}\left[(\bar{y}_{ij})_{j \leq N}^{i \leq B} \; , \; (\widetilde{y}_{ij})_{j \leq N}^{i \leq B} \right] , \tag{4}$$

where BCE$[\cdot, \cdot]$ stands for binary cross-entropy. As observed in [13], the sampling strategy used for points $\mathbf{x}_{ij}$ strongly impacts training effectiveness. We describe ours in the supplementary. In our experiments, we set $\delta = 0.1$, being the top and bottom garments normalized respectively into the upper and lower halves of the $[-1, 1]^3$ cube.

Gradient Loss. Given the same sample points as before, we take the gradient loss to be

$$\mathcal{L}_{grad} = \frac{1}{BN} \sum_{i,j} \| \mathbf{g}_{ij} - \widehat{\mathbf{g}_{ij}} \|_2^2 \; , \tag{5}$$

where $\mathbf{g}_{ij} = \nabla_{\mathbf{x}} y_{ij} \in \mathbb{R}^3$ is the ground-truth gradient of the i-th garment's UDF at $\mathbf{x}_{ij}$ and $\widehat{\mathbf{g}_{ij}} = \nabla_{\mathbf{x}} D_G(\mathbf{x}_{ij}, \mathbf{z}_i)$ the one of the predicted UDF, computed by backpropagation.

3.2. Garment Draping Network

We describe our approach to draping generic garments as opposed to specific ones and our self-supervised scheme. We assume that all garments are made of a single common fabric material, and we drape them in a quasi-static manner.

3.2.1 Draping Generic Garments

We rely on SMPL [24] to parameterize the body in terms of shape (β) and pose (θ) parameters. It uses Linear Blend Skinning to deform a body template. Since garments generally follow the pose of the underlying body, we extend the SMPL skinning procedure to the 3D volume around the body for garment draping. Given a point $\mathbf{x} \in \mathbb{R}^3$ in the garment space, its position $D(\mathbf{x}, \beta, \theta, \mathbf{z})$ after draping becomes

$$\begin{aligned} D(\mathbf{x}, \beta, \theta, \mathbf{z}) &= W(\mathbf{x}_{(\beta,\theta,\mathbf{z})}, \beta, \theta, \mathcal{W}(\mathbf{x})) \; , \\ \mathbf{x}_{(\beta,\theta,\mathbf{z})} &= \mathbf{x} + \Delta x(\mathbf{x}, \beta) + \Delta x_{\text{ref}}(\mathbf{x}, \beta, \theta, \mathbf{z}) \; , \\ \Delta x_{\text{ref}}(\mathbf{x}, \beta, \theta, \mathbf{z}) &= \mathcal{B}(\beta, \theta) \cdot \mathcal{M}(x, \mathbf{z}) \; , \end{aligned} \tag{6}$$

where $W(\cdot)$ is the SMPL skinning function, applied with blending weights $\mathcal{W}(\mathbf{x})$, over the point displaced by $\Delta x(\mathbf{x}, \beta)$ and $\Delta x_{\text{ref}}(\mathbf{x}, \beta, \theta, \mathbf{z})$. $\mathcal{W}(\mathbf{x})$ and $\Delta x(\mathbf{x}, \beta)$ are computed as in [19, 45]. However, they only give an initial deformation for garments that roughly fits the underlying body. To refine it, we introduce a new term, $\Delta x_{\text{ref}}(\mathbf{x}, \beta, \theta, \mathbf{z})$. It is a deformation field conditioned on body parameters β and θ, and on the garment latent code $\mathbf{z}$ from the generative network. Following the linear decomposition of displacements in SMPL, it is the composition of an embedding $\mathcal{B}(\beta, \theta) \in \mathbb{R}^{N_{\mathcal{B}}}$ of body parameters and a displacement matrix $\mathcal{M}(x, \mathbf{z}) \in \mathbb{R}^{N_{\mathcal{B}} \times 3}$ conditioned on $\mathbf{z}$. Being conditioned on the latent code $\mathbf{z}$, Δx_{ref} can deform different garments differently, unlike the methods of [5, 42]. The number of vertices does not need to be fixed, since displacements are predicted separately for each vertex.

Since we have distinct encodings for the top and bottom garments, for each one we train two MLPs ($\mathcal{B}$, $\mathcal{M}$) to predict Δx_{ref}. The other MLPs for $\mathcal{W}(\cdot)$ and $\Delta x(\cdot)$ are shared.

3.2.2 Self-Supervised Training

We first learn the weights of $\mathcal{W}(\cdot)$ and $\Delta x(\cdot)$ as in [19, 45], which does not require any annotation or simulation data but only the blending weights and shape displacements of SMPL. We then train our deformation fields Δx_{ref} in a fully self-supervised fashion by minimizing the physics-based losses introduced below. In this way, we completely eliminate the huge cost that extensive simulations would entail.

Top Garments. For upper body garments – shirts, t-shirts, vests, tank tops, etc. – the deformation field is trained using the loss from [42], expressed as

$$\mathcal{L}_{top} = \mathcal{L}_{strain} + \mathcal{L}_{bend} + \mathcal{L}_{gravity} + \mathcal{L}_{col} \; , \tag{7}$$

where $\mathcal{L}_{strain}$ is the membrane strain energy of the deformed garment, $\mathcal{L}_{bend}$ the bending energy caused by the folding of adjacent faces, $\mathcal{L}_{gravity}$ the gravitational potential energy, and $\mathcal{L}_{col}$ a penalty for collisions between body and garment. Unlike in [42], we only consider the quasi-static state after draping, that is, without acceleration.

Bottom Garments. Due to gravity, bottom garments, such as trousers, would drop onto the floors if we used only the loss terms of Eq. (7). We thus introduce an extra loss term to constrain the deformation of vertices around the waist and hips. The loss becomes

$$\mathcal{L}_{bottom} = \mathcal{L}_{strain} + \mathcal{L}_{bend} + \mathcal{L}_{gravity} + \mathcal{L}_{col} + \mathcal{L}_{pin},$$
$$\mathcal{L}_{pin} = \sum_{v \in V} |\Delta x_y|^2 + \lambda(|\Delta x_x|^2 + |\Delta x_z|^2) \ , \tag{8}$$

where V is the set of garment vertices whose closest body vertices are located in the region of the waist and hips. See supplementary material for details. The terms Δx_x, Δx_y and Δx_z are the deformations along the X, Y and Z axes, respectively. λ is a positive value smaller than 1 that penalizes deformations along the vertical direction (Y axis) and produces natural deformations along the other directions.

Top-Bottom Intersection. To ensure that the top and bottom garments do not intersect with each other when we drape them on the same body, we define a loss $\mathcal{L}_{IS}$ that ensures that when the top and the bottom garments overlap, the bottom garment vertices are closer to the body mesh than the top ones, which prevents them from intersecting – this is arbitrary, and the following could be formulated the other way around. To this end, we introduce an Intersection Solver (IS) network. It predicts a displacement correction Δx_{IS}, added only when draping bottom garments as

$$\tilde{\mathbf{x}}_{(\mathbf{z}_{top}, \mathbf{z}_{bot})} = \mathbf{x}_{(\mathbf{z}_{bot})} + \Delta x_{IS}(\mathbf{x}, \mathbf{z}_{top}, \mathbf{z}_{bot}) \ , \tag{9}$$

where we omit the dependency of $\tilde{\mathbf{x}}$, $\mathbf{x}$ and Δx_{IS} on the body parameters (β, θ) for simplicity. $\mathbf{z}_{top}$ and $\mathbf{z}_{bot}$ are the latent codes of the top and bottom garments, and $\mathbf{x}_{(\mathbf{z}_{bot})}$ is the input point displaced according to Eq. (6). The skinning function of Eq. (6) is then applied to $\tilde{\mathbf{x}}_{(\mathbf{z}_{top}, \mathbf{z}_{bot})}$ for draping. $\Delta x_{IS}(\cdot)$ is implemented as a simple MLP and trained with

$$\mathcal{L}_{IS} = \mathcal{L}_{bottom} + \mathcal{L}_{layer}, \tag{10}$$

where $\mathcal{L}_{layer}$ is a loss whose minimization requires the top and bottom garments to be separated from each other. We formulate it as

$$\mathcal{L}_{layer} = \sum_{v_B \in C} max(d_{bot}(v_B) - \gamma d_{top}(v_B), 0) \ , \tag{11}$$

where C is the set of body vertices covered by both the top and bottom garments, $d_{top}(\cdot)$ and $d_{bot}(\cdot)$ the distance to the top and the bottom garments respectively, and γ a positive value smaller than 1 (more details in the supplementary).

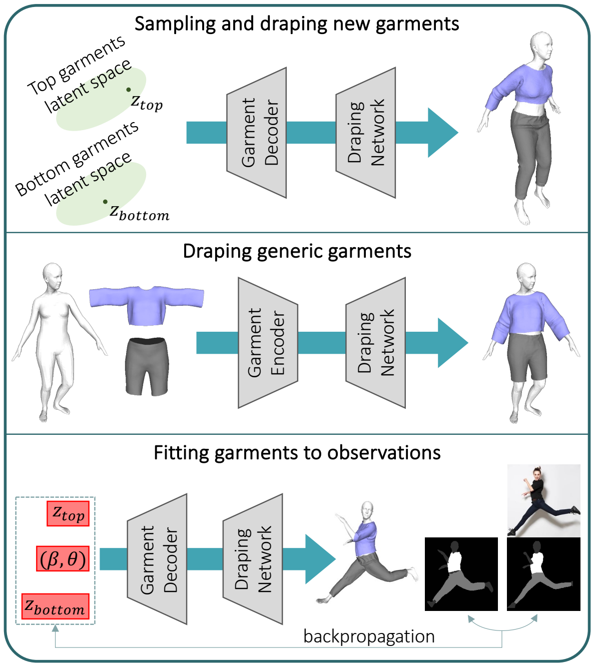

Figure 3. **Overview of** DrapeNet **applications. Top:** New garments can be sampled from the latent spaces of the generative networks, and deformed by the draping networks to fit to a given body. **Center:** The garment encoders and the draping networks form a general purpose framework to drape any garment with a single forward pass. **Bottom:** Being a differentiable parametric model, our framework can reconstruct 3D garments by fitting observations such as images or 3D scans. The red boxes indicate the parameters optimized in this process.

4. Experiments

We first describe our experimental setup and test DrapeNet for the different purposes depicted by Fig. 3. They include reconstructing different kinds of garments and editing them by manipulating their latent codes. We then gauge the draping network both qualitatively and quantitatively. Finally, we use DrapeNet to reconstruct garments from images and 3D scans.

4.1. Settings, Datasets and Metrics

Datasets. Both our generative and draping networks are trained with garments from CLOTH3D [4], a synthetic dataset that contains over 7K sequences of animated 3D humans parametrized used the SMPL model and wearing different garments. Each sequence comprises up to 300 frames and features garments coming from different templates. For training, we randomly selected 600 top garments (t-shirts, shirts, tank tops, etc.) and 300 bottom garments (both long and short trousers). Neither for the generative nor for the draping networks did we use the simulated deformations of the selected garments. Instead, we trained the networks using only garment meshes on average body shapes in T-pose.

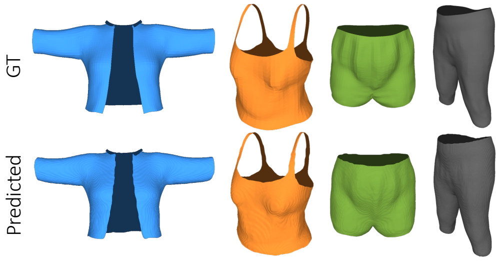

Figure 4. **Generative network: reconstruction of unseen garments in neutral pose/shape.** The latent codes are obtained with the garment encoder, then decoded into open surface meshes.

By contrast, for testing purposes, we selected random clothing items – 30 for top garments and 30 bottom ones – and considered *whole* simulated sequences.

Training. We train two different models for top and bottom garments, both for the generative and for the draping parts of our framework. First, the generative models are trained on the 600/300 neutral garments Then, with the generative networks weights frozen, we train the draping networks by following [42]: body poses θ are sampled randomly from the AMASS [26] dataset, and shapes β uniformly from $[-3, 3]^{10}$ at each step. The other hyperparameters are given in the supplementary material.

Metrics. We report the Euclidean distance (ED), interpenetration ratio between body and garment (B2G), and intersection between top and bottom garments (G2G). ED is computed between corresponding vertices of the considered meshes. B2G is the area ratio between the garment faces inside the body and the whole surface as in [19]. Since CLOTH3D exclusively features pairs of top/bottom garments with the bottom one closer to the body, G2G is computed by detecting faces of the bottom garment that are outside of the top one, and taking the area ratio between those and the overall bottom garment surface.

4.2. Garment Paramerization

We first test the encoding-decoding scheme of Sec. 3.1.

Encoding-Decoding Previously Unseen Garments. The generative network of Fig. 2 is designed to project garments into a latent space and to reconstruct them from the resulting latent vectors. In Fig. 4, we visualize reconstructed previously-unseen garments from CLOTH3D. The reconstructions are faithful to the input garments, including fine-grained details such as the shirt collar on the left or the shoulder straps of the tank top.

Semantic Manipulation of Latent Codes. Our framework enables us to edit a garment by manipulating its latent code. For the resulting edits to have a semantic meaning, we assigned binary labels corresponding to features of interest to 100 training garments. For instance, we labeled garments as having "short sleeves" (label = 0) or "long sleeves" (label = 1). Then, we fit a linear logistic regressor to the garment latent codes. After training, the regressor weights indicate which dimensions of the latent space control the feature of interest. To this end, we first apply min-max normalization to the absolute weight values and then zero out the ones below a certain threshold, empirically set to 0.5. The remaining non-zero weights indicate which dimensions of the latent codes should be increased or decreased to edit the studied feature. To create Fig. 5, we applied this simple procedure to control the sleeve length and the front opening for top garments along with the length for bottom garments. As can be seen from the figure, our latent representations give us the ability to edit a specific garment feature while leaving other aspects of the garment geometry unchanged.

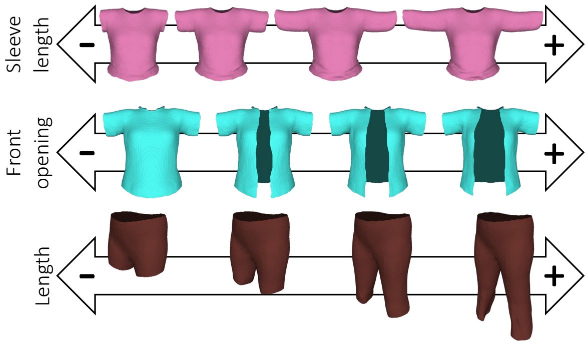

Figure 5. **Garment editing.** The latent codes produced by the garment encoder can be manipulated to edit specific features of the corresponding garments, without altering the overall geometry.

	DeePSD	DIG	Ours
ED-top (mm)	28.1	29.6	47.9
ED-bottom (mm)	18.3	20.0	27.3
B2G-top (%) ↓	7.2	1.8	**0.9**
B2G-bottom (%) ↓	3.4	0.8	**0.3**
G2G (%) ↓	2.0	4.0	**0.5**

Table 1. **Draping unseen garment meshes.** Comparison between DeePSD, DIG and our method, for top and bottom garments: Euclidean distance (ED), intersections with the body (B2G) and between garments (G2G) as ratio of intersection areas.

4.3. Garment Draping

We now turn to the evaluation of the draping network and compare its performance to those of DeePSD [6] or DIG [19], two *fully supervised* learning methods trained on CLOTH3D. DeePSD takes the point cloud of the garment mesh as input and predicts blending weights and pose displacements for each point; DIG drapes garments with a learned skinning field that can be applied to generic 3D points, but is similar for all garments. We chose those because, like `DrapeNet`, they both can deform garments of arbitrary geometry and topology.

Draping Unseen Meshes. We drape previously unseen garments on different bodies in random poses. We first encode the garments and use the resulting latent codes to condition the draping network, whose inference takes ~5ms.

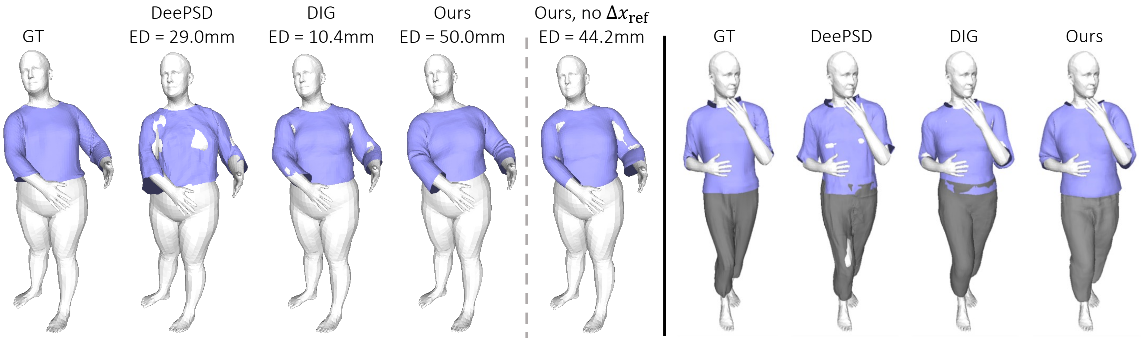

Figure 6. **Comparison between DeePSD, DIG and our method.** Ours is more realistic despite having the highest Euclidean distance (ED) error (**left**), and has less intersection between garments (**right**). **Left** also shows that Δx_{ref} is necessary for realistic deformations.

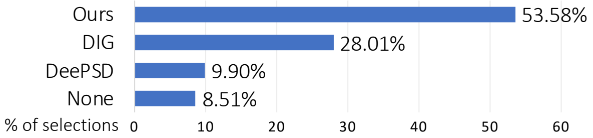

Figure 7. **Human evaluation of draping results.** When shown draping results of our method, DIG and DeePSD, evaluators selected ours as the most realistic one in more than half of the cases. *None* refers to the case when they had no clear preference.

We provide qualitative results in Fig. 6 and report quantitative ones in Tab. 1. Despite being completely self-supervised, DrapeNet delivers the lowest ratio of body-garment interpenetrations (B2G) for both top and bottom garments and the least intersections between them (G2G).

However, DrapeNet also yields higher ED values, which makes sense because there is more than one way to satisfy the physical constraints and to achieve realism. Hence, in the absence of explicit supervision, there is no reason for the answer picked by DrapeNet to be exactly the same as the one picked by the simulator. In fact, as argued in [5] and illustrated by Fig. 6, which is representative in terms of ED, a low ED value does not necessarily correspond to a realistic draping. To confirm this, we conducted a human evaluation study by sharing a link to a website on friends groupchats. We gave no further instructions or details besides those given on the site and reproduced in the supplementary material. The website displays 3 drapings of the same garment over the same posed body, one computed using our method and the others using the other two. The users were asked to select which one of the three seemed more realistic and more pleasant, with a fourth potential response being "none of them". We obtained feedback from 187 different people. A total of 1258 individual examples were rated and we collected 3738 user opinions. In other words, each user expressed 20 opinions on average. The chart in Fig. 7 shows that our method was selected more than 50% of the times, with a large gap over the second best, DIG [19], selected less than 30% of the time. This result confirms that DrapeNet can drape garments with better perceptual quality than the competing methods.

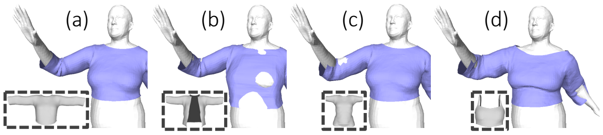

Figure 8. **Switching input latent codes of the draping network.** Draping the same shirt by conditioning the draping network with **(a)** the corresponding latent code, **(b)** the code of an open vest, **(c)** of a t-shirt and **(d)** of a tank top. Gray meshes in dashed boxed are the garments corresponding to the input latent codes.

Ablation Study. In Fig. 8, we show what happens when the draping network is conditioned with a latent code of a garment that does not match the input one. This creates unnatural deformations on the front when using the code of a shirt with a front opening to deform a shirt without an opening. Similarly, the sleeves penetrate the arms when conditioning with the code of a short sleeves shirt. This demonstrates that the draping network truly exploits the latent codes to predict garment-dependent deformation fields.

In Fig. 6 **left** we show that removing our novel displacement term $\Delta x_{\text{ref}}(\cdot)$ from Eq. (6) leads to unrealistic results.

We also ablate the influence of our Intersection Solver and observe that G2G increases from 0.5% to 1.1% without it. This demonstrates the effectiveness of this component at reducing collisions between top and bottom garments.

4.4. Fitting Observations

Since our method is end-to-end differentiable, it can be used to reconstruct 3D models of people and their garments from partial observations, such as 2D images and 3D scans.

Fitting Images. Given an image of a clothed person, we use the algorithm of [57, 58] to get initial estimates for the body parameters (β, θ) and a segmentation mask $\mathbf{S}$. Then, starting with the mean of the learned codes $\mathbf{z}$, we reconstruct a mesh for the body and its garments by minimizing

$$\begin{aligned} L(\beta,\theta,\mathbf{z}) &= L_{\text{IoU}}(R(D(\mathbf{G},\beta,\theta,\mathbf{z}),\text{SMPL}(\beta,\theta)),\ \mathbf{S})\ , \\ \mathbf{G} &= \text{MeshUDF}(D_G(\mathbf{z}))\ , \end{aligned} \tag{12}$$

w.r.t. $\mathbf{z}$, β and θ, where L_{IoU} is the IoU loss [20] in pixel space penalizing discrepancies between 2D masks, $R(\cdot)$ is

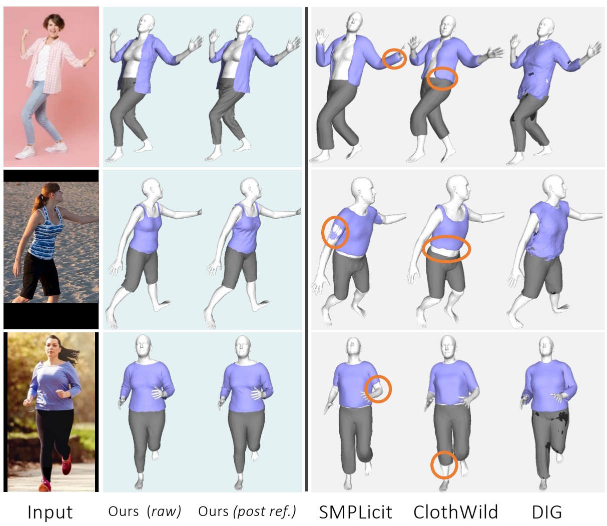

Figure 9. **Recovering garments and bodies from images.** From left to right we show the input image and the 3D models recovered with our method (without and with post-refinement), and competitors methods: SMPLicit [10], ClothWild [29], DIG [19].

a differentiable mesh renderer [40], and $\mathbf{G}$ is the set of vertices of the garment mesh reconstructed with our garment decoder using $\mathbf{z}$. $D(\cdot)$ and $\text{SMPL}(\cdot)$ are the garment and body skinning functions defined in Eq. (6) and in [24], respectively. To ensure pose plausibility, θ is constrained by an adversarial pose prior [11].

For the sake of simplicity, Eq. (12) formulates the reconstruction of a single garment $\mathbf{G}$. In practice, we extend this formulation to both the top and the bottom garments shown in the target image. Fig. 9 depicts the results of minimizing this loss. It outperforms the state-of-the-art methods SMPLicit [10], ClothWild [29] and DIG [19]. The garments we recover follow the ones in the input image with higher fidelity and visual quality, without interpenetration between the body and the garments or between the two garments.

After this optimization, we can further refine the result by minimizing the physics-based objectives of Eq. (7) w.r.t. the per-vertex displacements of the reconstructed garments, as opposed to w.r.t. the latent vectors. We describe this procedure in the supplementary material. As shown in the third column of Fig. 9, this further boosts the realism of the reconstructed garments. Note that this refinement is feasible thanks to the open surface representation allowed by our UDF model. Applying these physically inspired losses to an inflated garment, as produced by SMPLicit, ClothWild and DIG, yields poor results with many self-intersections, as shown in the supplementary material.

Fitting 3D scans. Given a 3D scan of a clothed person and segmentation information, we apply a strategy similar to the one presented above and minimize

$$L(\beta,\theta,\mathbf{z}) = d(D(\mathbf{G},\beta,\theta,\mathbf{z}),\ \mathbf{S_G}) + \vec{d}(\text{SMPL}(\beta,\theta),\ \mathbf{S_B}), \quad (13)$$

w.r.t. $\mathbf{z}$, β and θ, where $\mathbf{S_G}$ and $\mathbf{S_B}$ denote the segmented garment and body scan points, and $d(a,b)$ and $\vec{d}(a,b)$ are the bidirectional and the one-directional Chamfer distance from b to a. Similarly to Eq. (12), we apply Eq. (13) to recover both the top and bottom garments. Fig. 10 shows our fitting results for some scans of the SIZER dataset [50]. The recovered 3D models closely match the input scans. Moreover, we can also apply a post-refinement procedure similar to the one described above, by minimizing both the physics-based losses from Eq. (7) and the Chamfer distance to the input scan w.r.t. the 3D coordinates of the vertices of the reconstructed models. This leads to even more realistic results, with fine wrinkles aligning to the input scans.

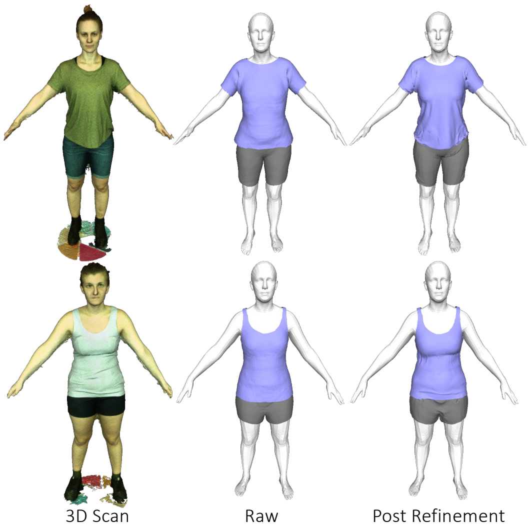

Figure 10. **Recovering garments and bodies from 3D scans.** We show 3D models recovered with our method from scans of the SIZER dataset [50]. *Raw* indicates the model recovered with Eq. (13) from the 3D scan. *Post Refinement* refers to the models further refined with the physics-based losses.

5. Conclusion

We have shown that physics-based self-supervision can be leveraged to learn a single parameterization for many different garments to be draped on human bodies in arbitrary poses. Our approach relies on UDFs to represent garment surfaces and on a displacement field to drape them, which enables us to handle a continuous manifold of garments without restrictions on their topology. Our whole pipeline is differentiable, which makes it suitable for solving inverse problems and for modeling clothed people from image data.

Future work will focus on modeling dynamic poses instead of only static ones. This is of particular relevance for loose clothes, where our reliance on the SMPL skinning prior should be relaxed. Moreover, we will investigate replacing our current global latent code by a set of local ones to yield finer-grained control both for garment editing and draping.

Acknowledgement. This project was supported in part by the Swiss National Science Foundation.

References

[1] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman. Controlling Neural Level Sets. In *Advances in Neural Information Processing Systems*, 2019. 2

[2] M. Atzmon and Y. Lipman. SALD: Sign Agnostic Learning with Derivatives. In *International Conference on Learning Representations*, 2020. 4

[3] D. Baraff and A. Witkin. Large Steps in Cloth Simulation. In *ACM SIGGRAPH*, pages 43–54, 1998. 1, 2

[4] H. Bertiche, M. Madadi, and S. Escalera. CLOTH3D: Clothed 3D Humans. In *European Conference on Computer Vision*, pages 344–359, 2020. 5

[5] H. Bertiche, M. Madadi, and S. Escalera. PBNS: Physically Based Neural Simulation for Unsupervised Garment Pose Space Deformation. *ACM Transactions on Graphics*, 2021. 2, 3, 4, 7

[6] H. Bertiche, M. Madadi, E. Tylson, and S. Escalera. DeePSD: Automatic Deep Skinning and Pose Space Deformation for 3D Garment Animation. In *International Conference on Computer Vision*, 2021. 2, 6

[7] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-Garment Net: Learning to Dress 3D People from Images. In *International Conference on Computer Vision*, 2019. 2

[8] T. Buffet, D. Rohmer, L. Barthe, L. Boissieux, and M-P. Cani. Implicit untangling: A robust solution for modeling layered clothing. *ACM Transactions on Graphics*, 38(4):1–12, 2019. 2

[9] J. Chibane, A. Mir, and G. Pons-Moll. Neural Unsigned Distance Fields for Implicit Function Learning. In *Advances in Neural Information Processing Systems*, 2020. 2

[10] E. Corona, A. Pumarola, G. Alenya, G. Pons-Moll, and F. Moreno-Noguer. Smplicit: Topology-Aware Generative Model for Clothed People. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2, 8

[11] A. Davydov, A. Remizova, V. Constantin, S. Honari, M. Salzmann, and P. Fua. Adversarial Parametric Pose Prior. In *Conference on Computer Vision and Pattern Recognition*, 2022. 8

[12] M. Designer, 2018. https://www.marvelousdesigner.com. 1

[13] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas. Curriculum DeepSDF. In *European Conference on Computer Vision*, 2020. 4

[14] B. Guillard, F. Stella, and P. Fua. MeshUDF: Fast and Differentiable Meshing of Unsigned Distance Field Networks. In *European Conference on Computer Vision*, 2022. 2, 3, 4

[15] E. Gundogdu, V. Constantin, S. Parashar, A. Seifoddini, M. Dang, M. Salzmann, and P. Fua. 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. 1, 2

[16] E. Gundogdu, V. Constantin, A. Seifoddini, M. Dang, M. Salzmann, and P. Fua. Garnet: A Two-Stream Network for Fast and Accurate 3D Cloth Draping. In *International Conference on Computer Vision*, 2019. 1

[17] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao. Bcnet: Learning body and cloth shape from a single image. In *European Conference on Computer Vision*, 2020. 2

[18] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient Implementation of Marching Cubes' Cases with Topological Guarantees. In *Journal of Graphics Tools*, 2003. 2

[19] R. Li, B. Guillard, E. Remelli, and P. Fua. DIG: Draping Implicit Garment over the Human Body. In *Asian Conference on Computer Vision*, 2022. 2, 3, 4, 6, 7, 8

[20] R. Li, M. Zheng, S. Karanam, T. Chen, and Z. Wu. Everybody Is Unique: Towards Unbiased Human Mesh Recovery. In *British Machine Vision Conference*, 2021. 7

[21] Y. Li, M. Habermann, B. Thomaszewski, S. Coros, T. Beeler, and C. Theobalt. Deep physics-aware inference of cloth deformation for monocular human performance capture. In *International Conference on 3D Vision*, 2021. 2

[22] J. Liang, M. Lin, and V. Koltun. Differentiable Cloth Simulation for Inverse Problems. In *Advances in Neural Information Processing Systems*, 2019. 2

[23] T. Liu, S. Bouaziz, and L. Kavan. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Transactions on Graphics*, 2017. 1

[24] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M.J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM SIGGRAPH Asia*, 34(6), 2015. 3, 4, 8

[25] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black. Learning to Dress 3D People in Generative Clothing. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1

[26] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*, pages 5442–5451, 2019. 6

[27] I. Mehta, M. Chandraker, and R. Ramamoorthi. A Level Set Theory for Neural Implicit Evolution under Explicit Flows. In *European Conference on Computer Vision*, 2022. 2

[28] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[29] G. Moon, H. Nam, T. Shiratori, and K.M. Lee. 3d clothed human reconstruction in the wild. In *European Conference on Computer Vision*, 2022. 8

[30] R. Narain, T. Pfaff, and J.F. O'Brien. Folding and crumpling adaptive sheets. *ACM Transactions on Graphics*, 2013. 2

[31] R. Narain, A. Samii, and J.F. O'brien. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics*, 2012. 1, 2

[32] Nvidia. Nvcloth, 2018. 1

[33] Nvidia. NVIDIA Flex, 2018. https://developer.nvidia.com/flex. 1

[34] X. Pan, J. Mai, X. Jiang, D. Tang, J. Li, T. Shao, K. Zhou, X. Jin, and D. Manocha. Predicting loose-fitting garment deformations using bone-driven motion networks. In *ACM SIGGRAPH*, 2022. 2

[35] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove. Deepsdf: Learning Continuous Signed Distance

Functions for Shape Representation. In *Conference on Computer Vision and Pattern Recognition*, 2019. 2

[36] C. Patel, Z. Liao, and G. Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[37] G. Pons-Moll, S. Pujades, S. Hu, and M.J. Black. Clothcap: Seamless 4D Clothing Capture and Retargeting. *ACM SIGGRAPH*, 36(4):731–7315, July 2017. 1

[38] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation*. 1997. 1

[39] Xavier Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, 1995. 1

[40] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. PyTorch3D. https://github.com/facebookresearch/pytorch3d, 2020. 8

[41] E. Remelli, A. Lukoianov, S. Richter, B. Guillard, T. Bagautdinov, P. Baque, and P. Fua. Meshsdf: Differentiable Iso-Surface Extraction. In *Advances in Neural Information Processing Systems*, 2020. 2

[42] I. Santesteban, M.A. Otaduy, and D. Casas. SNUG: Self-Supervised Neural Dynamic Garments. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 4, 5, 6

[43] I. Santesteban, M.A. Otaduy, N. Thuerey, and D. Casas. Ulnef: Untangled layered neural fields for mix-and-match virtual try-on. In *Advances in Neural Information Processing Systems*, 2022. 2

[44] I. Santesteban, M. A. Otaduy, and D. Casas. Learning-Based Animation of Clothing for Virtual Try-On. *Computer Graphics Forum (Proc. of Eurographics)*, 33(2), 2019. 1

[45] I. Santesteban, N. Thuerey, M. A. Otaduy, and D. Casas. Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. In *Conference on Computer Vision and Pattern Recognition*, 2021. 2, 4

[46] Y. Shen, J. Liang, and M.C. Lin. Gan-based garment generation using sewing pattern images. In *European Conference on Computer Vision*, 2020. 1

[47] Optitext Fashion Design Software, 2018. https://optitex.com/. 1

[48] Tongkui Su, Yan Zhang, Yu Zhou, Yao Yu, and Sidan Du. GPU-based Real-time Cloth Simulation for Virtual Try-on. In *Pacific Conference on Computer Graphics and Applications*, 2018. 1, 2

[49] M. Tang, R. Tong, R. Narain, C. Meng, and D. Manocha. A GPU-based streaming algorithm for high-resolution cloth simulation. In *Computer Graphics Forum*, 2013. 1

[50] G. Tiwari, B. L. Bhatnagar, T. Tung, and G. Pons-Moll. Sizer: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing. In *European Conference on Computer Vision*, 2020. 1, 2, 8

[51] T. Vassilev, B. Spanlang, and Y. Chrysanthou. Fast cloth animation on walking avatars. In *Computer Graphics Forum*, 2001. 1

[52] R. Venkatesh, T. Karmali, S. Sharma, A. Ghosh, R. V. Babu, L. A. Jeni, and M. Singh. Deep Implicit Surface Point Prediction Networks. In *International Conference on Computer Vision*, 2021. 2

[53] R. Vidaurre, I. Santesteban, E. Garces, and D. Casas. Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. In *Computer Graphics Forum*, 2020. 1

[54] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A.C. Courville. Modulating Early Visual Processing by Language. In *Advances in Neural Information Processing Systems*, 2017. 4

[55] T. Y. Wang, D. Ceylan, J. Popovic, and N. J. Mitra. Learning a Shared Shape Space for Multimodal Garment Design. In *ACM SIGGRAPH Asia*, 2018. 1

[56] Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. Bronstein, and J.M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. In *ACM Transactions on Graphics*, 2019.

[57] Yu Y. Rong, T. Shiratori, and H. Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. In *International Conference on Computer Vision Workshops*, 2021. 7

[58] L. Yang, Q. Song, Z. Wang, M. Hu, C. Liu, X. Xin, W. Jia, and S. Xu. Renovating parsing R-CNN for accurate multiple human parsing. In *European Conference on Computer Vision*, 2020. 7

[59] I. Zakharkin, K. Mazur, A. Grigorev, and V. Lempitsky. Point-based modeling of human clothing. In *International Conference on Computer Vision*, 2021. 2

[60] C. Zeller. Cloth simulation on the gpu. In *ACM SIGGRAPH*. 2005. 1

[61] F. Zhao, W. Wang, S. Liao, and L. Shao. Learning Anchored Unsigned Distance Functions with Gradient Direction Alignment for Single-View Garment Reconstruction. In *International Conference on Computer Vision*, 2021. 2, 4