CLOTH3D: Clothed 3D Humans

Hugo Bertiche[1,2][0000−0002−6632−1902], Meysam Madadi[1,2][0000−0002−7384−5712], and Sergio Escalera[1,2][0000−0003−0617−8873]

[1] Universitat de Barcelona, Spain
[2] Computer Vision Center, Spain
hugo_bertiche@hotmail.com

Fig. 1: Left: CLOTH3D[3] is the first big scale dataset of animated clothed humans. It contains thousands of different outfits and subjects, high variability of poses and rich cloth dynamics. Right: generated 3D garments with proposed GCVAE.

Abstract. We present CLOTH3D, the first big scale synthetic dataset of 3D clothed human sequences. CLOTH3D contains a large variability on garment type, topology, shape, size, tightness and fabric. Clothes are simulated on top of thousands of different pose sequences and body shapes, generating realistic cloth dynamics. We provide the dataset with a generative model for cloth generation. We propose a Conditional Variational Auto-Encoder (CVAE) based on graph convolutions (GCVAE) to learn garment latent spaces. This allows for realistic generation of 3D garments on top of SMPL model for any pose and shape.

Keywords: 3D · Human · Garment · Cloth · Dataset · Generative model

1 Introduction

The modelling, recovery and generation of 3D clothes will allow for enhanced virtual try-ons experience, reducing designers and animators workload, or understanding of physics simulations through deep learning, just to mention a few.

[3] http://chalearnlap.cvc.uab.es/dataset/38/description/

Dataset	3DPW[18]	BUFF[35]	Untitled[29]	3DPeople[24]	TailorNet[21]	CLOTH3D
Resolution	2.5cm	0.4cm	1cm	-[1]	1cm	1cm
Missing	x	✓	x	x	x	x
Dynamics	x	✓	x	x	x	✓
Garments	18[2]	10∼ 20	3[3]	High[4]	20	11.3K
Fabrics	x	x	x	x	x	✓
Poses[5]	Low	Low	Very low	Low	1782	High
Subjects	18[2]	6	2K	80	9	8.5K
Layered	x	x	✓	-[1]	✓	✓
#samples	51k	11K	24K	2.5M	55.8k	2.1M
Type	Real	Real	Synth.	Synth.	Synth.	Synth.
RGB	✓	x	✓	✓	x	x
GT error	26mm	1.5-3mm	None	None	None	None

Table 1: CLOTH3D vs. available 3D cloth datasets. [1]: 3D data includes depth, normal and scene flow maps, but not 3D models. [2]: 3DPW contains 18 clothed models that can be shaped as SMPL. [3]: garments of [29] are shaped to different sizes. [4]: Garment variability not specified, nonetheless, authors propose a generation pipeline that can modify template garments into many different sizes. [5]: poses are strongly related to number of frames, and in [29] most samples share the same static pose.

However, current literature in the modelling, recovery and generation of clothes is almost focused on 2D data [8, 13, 23, 27]. This is because of two factors. First, deep learning approaches are data-hungry, and nowadays not enough 3D cloth data is available (see Tab. 1). Second, garments present a huge variability in terms of shape, sizes, topologies, fabrics, or textures, among others, increasing the complexity of representative 3D garment generation.

One could define three main strategies in order to produce data of 3D dressed humans: 3D scans, 3D-from-RGB, and synthetic generation. In the case of 3D scans, they are costly, and at most they can produce a single mesh (human + garments). Alternatively, datasets that infer 3D geometry of clothes from RGB images are inaccurate and cannot properly model cloth dynamics. Finally, synthetic data is easy to generate and is ground truth error free. Synthetic data has proved to be helpful to train deep learning models to be used in real applications [20, 25, 28].

In this work, we present CLOTH3D, the first synthetic dataset composed of thousands of sequences of humans dressed with high resolution 3D clothes, see Fig.1. CLOTH3D is unique in terms of garment, shape, and pose variability, including more than 2 million 3D samples. We developed a generation pipeline that creates a unique outfit for each sequence in terms of garment type, topology, shape, size, tightness and fabric. While other datasets contain just a few different garments, ours has thousands of different ones. On Tab. 1 we summarize features of existing datasets and CLOTH3D.

Additionally, we provide a baseline model able to generate dressed human models. Similar to [2, 17, 32] we encode garments as offsets connecting skin to cloth, using SMPL[15] as human body model. This yields an homogeneous dimensionality on the data. As in [22], we use a segmentation mask to extract the garment by removing body vertices. In our case, the mask is predicted by the network. We propose a Conditional Variational Auto-Encoder (CVAE) based on graph convolutions [6, 7, 17, 19, 31, 34] (GCVAE) to learn garment latent spaces. This later allows for the generation of 3D garments on top of SMPL model for any pose and shape (right on Fig.1).

2 Related Work

3D Garment Datasets. Current literature on 3D garment lacks on large public available datasets. One strategy to capture 3D data is through **3D scans**. The BUFF dataset [35] provides high resolution 3D scans, but few number of subjects, poses and garments. Furthermore, scanning techniques cannot provide layered models (one mesh for the body and one for each garment) and often one can find regions occluded at scanning time, meaning missing vertices or corrupted shapes. The work of [22] proposed a methodology to segment scans to obtain layered models. Authors of [33] combined 3D scans with cloth simulation fitting at each frame to deal with missing vertices. Similarly, [5] provided a dataset from 3D scans. However, the amount of samples is in the order of a few hundreds. The 3DPW dataset [18] is not focused on garments, but rather on pose and shape in-the-wild. The authors proposed a modified SMPL **parameterized model** for each outfit (18 clothed models), which, as SMPL, can be shaped and posed. Nevertheless, resolution is low and posing is through rigid rotations. Therefore, cloth dynamics are not represented. The dataset of [29] is synthetically created through **physics simulation**, with three different garment types: tshirt, skirt and kimono. They propose an automatic garment resizing based on real patterns, but provide only static samples on few different poses. The work of [21] also includes a synthetic dataset obtained through simulation of 20 combinations of different garment styles and body shapes into 1782 static poses. Finally, 3DPeople dataset [24] is the most comparable to ours in terms of scale, but has significant differences w.r.t. CLOTH3D. On one hand, this dataset has been designed specifically for computer vision. Data are given as **multi-view images** (RGB, depth, normal and scene flow), there are no 3D models. On the other hand, the garments are rigged models, so there is no proper cloth dynamics. And lastly, source pose data is sparse, 70 pose sequences with an average length of 110 frames. Our CLOTH3D dataset aims to overcome previous datasets issues. We automatically generate garments to obtain a huge variability on garment type, topology, shape, size, tightness and fabric. Afterwards, we simulate clothes on top of thousands of different pose sequences and body shapes. Tab.1 shows a comparison of features for existing datasets and ours. In CLOTH3D we focus on sample variability (garments, poses, shapes), containing realistic cloth dynamics. 3DPW and 3DPeople sequences are based on rotations on rigged mod-

els, datasets of [21, 29] contain static poses only, and BUFF has very few and short sequences. Moreover, none other provides metadata about fabrics, which has a strong influence on cloth behaviour. Similarly, the scarcity of these datasets implies low variability on garments, poses and subjects. Finally, note how only synthetic datasets provide with layered models and have no annotation error.

3D Garment Generation. Current works in 3D clothing focus on the generation of dressed humans. We split related work into non-deep and deep-learning approaches. Regarding **non-deep learning**, the authors of [10] proposed a data-driven model that learns deformations from template garment to garment fitted to the human body, shaped and posed. They factorize deformations into shape-dependant and pose-dependant by training on rest pose data first, and later on posed bodies. Transformations are learnt per triangle, and thus it yields inconsistent meshes that need to be reconstructed. The data-driven model of [22] is able to recover and retarget garments from 4D scan sequences relying on masks to separate body and cloth. Authors propose an energy optimization process to identify underlying body shape and garment geometry, later, cloth displacements w.r.t. body are computed and applied to new body shapes. This means information such as wrinkles is "copied" to new bodies, which produces valid samples but cannot properly generate its variability. Regarding **deep learning** strategies, the work of [11] deals with body and garments as different point clouds through different streams of a network, which are later fused. They also use skin-cloth correspondences for computing local-features and losses through nearest neighbour. The works of [2, 17, 21, 32] consider encoding clothes as offsets from SMPL body model with different goals. In [17] authors propose a combination of graph VAE and GAN to model SMPL offsets into clothing. Similarly, in [21], authors propose encoding garments as SMPL offsets and topology as a subset of SMPL vertices, later, they learn two models for low and high frequency details which effectively generate realistic wrinkles on the garments. In [29, 32] a PCA decomposition is used to reduce clothing space. In [3, 12], authors register garments to low resolution meshes (garment templates and SMPL respectively), to later use UV normal maps to represent high-frequency cloth details (wrinkles). Authors of [26] propose learning Pose Space Deformation models for template garments by training deep models instead of SVD (as SMPL). The work of [30] presents a template garment autoencoder where latent spaces are disentangled into motion and static properties to realistically interpolate into 3D keyframes. Similar to previous approaches, our proposed methodology also encodes clothes as SMPL offsets. Nevertheless, the assumption that garments follow body topology does not hold for skirts and dresses. In this sense, we propose a novel body topology specific for those cases. Additionally, our model predicts garment mask along offsets to generate layered models.

3 Dataset

CLOTH3D is the first big scale dataset of 3D clothed humans. The dataset is composed of 3D sequences of animated human bodies wearing different garments.

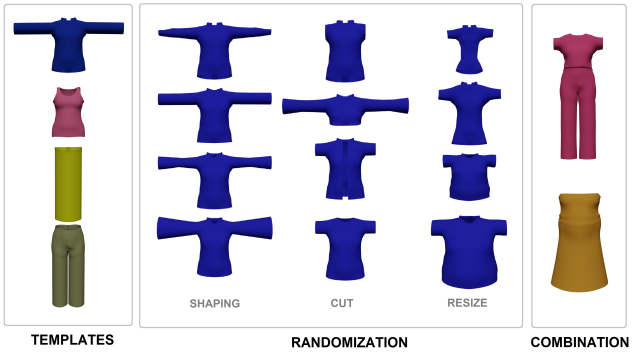

Fig. 2: Unique outfit generation pipeline. First, one upper-body and lower-body garment template is selected. Then, garments are individually shaped, cut and resized. Finally, garments might be combined into a single one.

Fig. 1 depicts a sequence (first row) and randomly sampled frames from different sequences. Samples are layered, meaning each garment and body are represented by different 3D meshes. Garments are automatically generated for each sequence with randomized shape, tightness, topology and fabric, and resized to target human shape. This process yields a unique outfit for each sequence. It contains over 7000 non-overlapping sequences of 300 frames each at 30fps, yielding a total of 2.1M samples. As seen in Tab. 1, garment and pose variability is scarce in available datasets, and CLOTH3D aims to fill that gap. To ensure garment type balance, given that females present higher garment variability, we balance gender as 2:1 (female:male). Finally, for validation purposes, we split the data in 80% sequences as training and 20% as test. Splitting by sequences ensures no garment, shape or pose is repeated in training and test.

The data generation pipeline starts with sequences of human bodies in 3D. Human pose data source is [1], later transformed to volumetric bodies through SMPL [15]. These sequences might present body self-collisions which will hinder cloth simulation, not only on affected regions, but also in global garment dynamics. We automatically solve collisions or reject these samples. Human generation process is described in subsec. 3.1. Later, we generate unique outfits for each sequence. We start from a few template meshes which are randomly shaped, cut and resized to generate a unique pair of garments for each sample, with the possibility to be combined into a single full-body garment. Fig. 2 shows the generation process, which is also detailed in subsec. 3.2. Finally, once human sequence and outfit are done, we use a physics based simulation to obtain the garment 3D sequences. Simulation details are described in subsec. 3.3.

3.1 Human 3D Sequences

SMPL. It is a parametric human body model which takes as input shape $\beta \in \mathbb{R}^{10}$ and pose $\theta \in \mathbb{R}^{24\times 3}$ to generate the corresponding mesh with 6890

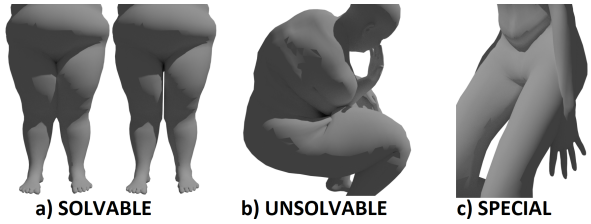

Fig. 3: Types of self-collision: a) collided vertices can be linearly separated with the aid of a body part segmentation, b) no trivial solution, we reject this kind of sample, c) correct simulation might be possible if forearm is removed.

vertices. We use this model to generate animated human 3D sequences. We refer to [16] for SMPL details. To generate animated bodies, we need a source of valid sequences of SMPL pose parameters $\theta \in \mathbb{R}^{f\times 24\times 3}$. We take such data from the work of [28], where pose is inferred from CMU MoCap data [1] following the methodology proposed at [14]. These pose data come from around 2600 sequences of 23 different actions (dancing, playing, running, walking, jumping, climbing, etc.) performed by over 100 different subjects. SMPL shape deformations are linearly modeled through PCA. To obtain a balanced dataset we uniformly sample shape within range $[-3, 3]$ for each sequence.

Self-collision. Body collides with itself for certain combinations of pose and shape parameters. Intersection volumes create regions where simulated repel forces are inconsistent, corrupting global cloth dynamics. We classify these collisions in three generic cases. Solvable Fig.3(a): small intersection volumes near joints, specially armpits and crotch. We use SMPL body parts segmentation to linearly separate the collided vertices to permit a correct simulation. Separation space is 4mm so that a folded cloth can fit. Unsolvable Fig.3(b): big intersection volumes or incompatible intersections (e.g.: arm vs. leg). We reject or re-simulate with thinner body. Special cases Fig.3(c): removing hands, forearms or arms for short-sleeved upper-body and lower-body garments significantly increases the amount of valid samples. This requires manual supervision. Self-collision solution is not stored, hence, if collided vertices change significantly, garments might present interpenetration w.r.t. unsolved body. Only small intersected volumes are corrected and the rest are rejected (or simulated with thinner body). The goal of self-collision solving is to avoid invalid cloth dynamics. Accurate, realistic solving of soft-body self-collision is out of the scope of this work.

3.2 Garment Generation

Garment Templates. Generation starts with a few template garments for each gender. Garments can be classified in upper-body and lower-body. Lower-body can be further split into trousers and skirts. These three categories, and combinations between them, encompass almost any day-to-day garment. Template

garments have been manually created by designers from real patterns and are: t-shirt, top, trousers and skirt.

Shaping. On sleeves, legs and skirt, we find a significant shape variability. It is possible to define them as cylinders of variable width around certain axes: along arms for sleeves, legs for trousers and vertical body axis for skirt. For sleeves and legs, width will be constant or decreasing while moving towards wrist/ankle, and beyond a randomly sampled point along its axis, it might start increasing (widening). For skirts, width always increases, from waist to bottom. Rate of width decrease/increase is uniformly sampled within ranges empirically set per garment. More formally:

$$W(x) = \alpha_1 x + \alpha_2 \max(0, x - x_{offset}) + W_0, \tag{1}$$

where x is position along axis (0 at shoulder/hips), $W(x)$ is width at position x, W_0 is width at $x = 0$, x_{offset} is a uniformly sampled point along the axis and α_1 and α_2 are constants empirically defined for each garment. For t-shirts and trousers, $\alpha_1 < 0 < \alpha_2$. For skirts, $\alpha_1 > \alpha_2 = 0$.

Cut. Template garments cover most of the body (long sleeves, legs and skirt). At this generation step, garments are cut to increase variability on length and topology. Cuts are along arms, legs and torso. Plus, upper-body garments have specific cuts to generate different types of garments (e.g., t-shirt, shirt, polo).

Resizing. Garments are resized to random body shapes. It is safe to assume that size variability on garments is similar to body shape variability. Following this reasoning, SMPL shape displacements are transferred to garments by nearest neighbour. Nevertheless, this process is noisy and human body details are transferred to garment. To address these issues, an iterative Laplacian smoothing is applied to shape displacements, removing noise and filtering high frequency body details, while preserving the geometry of the original garment. On SMPL, first and second shape parameters correspond to global human size and overall fatness. Knowing this, garments are resized to a different target shape. This new shape has two offsets at first and second parameters, the garment tightness $\gamma \in \mathbb{R}^2$. These offsets on garment resizing will generate loose or tight variability. As tighter garments present less dynamics and complexity, we bias the generator towards loose clothes by sampling tightness on the range $[-1.5, 0.5]$.

Jumpsuits and Dresses. Full-body garments can be generated by combining upper-body and lower-body garments. After generating the clothes individually, a final step automatically sews them together.

3.3 Simulation

Cloth simulation is performed on Blender, an open source 3D creation suite. Blender's cloth physics, as it is in version 2.8, has been implemented with state-of-the-art algorithms based on mass-spring model. The simulation performs $420-600$ steps per second, depending on the complexity of the garment.

Fabrics. Changing the parameters of the mass-spring model allows simulation of different fabrics. Blender provides different presets for *cotton*, *leather*, *silk*

Walk	Animal	Fight	Jump	Run	Sing	Wait	Swim	Story	Sports	Dance	Yoga	Spin
27.49%	10.79%	4.38%	2.78%	2.49%	2.38%	2.31%	1.97%	1.70%	1.63%	1.37%	1.01%	0.90%

Exercise	Climb	Carry	Stand	Wash	Balancing	Trick	Sit	Interact	Drink	Pose	Others
0.84%	0.71%	0.67%	0.66%	0.63%	0.54%	0.51%	0.28%	0.20%	0.14%	0.14%	33.48%

Table 2: CLOTH3D statistics per action label.

and *denim*, among others. These four fabrics have been used for the creation of the dataset. Upper-body garments might be cotton or silk, while the rest of the garment types can be any of those fabrics. Different fabrics produce different dynamics and wrinkles on simulation time.

Elastics. At simulation time, sleeves and legs have a 50% chance each of presenting an elastic behaviour at their ends, also at waist on full-body garments.

3.4 Additional dataset statistics

Tab.2 shows the CLOTH3D statistics in terms of action labels by grouping them into generic categories. Note that original data action label is very heterogeneous, specific and incomplete. These labels are gathered from CMU MoCap dataset. We observe a high density on *Walk*, but it is important to note that this gathers many different sub-actions (walk backwards, zombie walk, walk stealthily, ...) as many other action labels do. Additionally, most of these actions were performed by different subjects, which implies an increase in intra-class variability. The label 'others' contains all action labels that cannot be included in any of the categories plus all the missing action labels.

4 Dressed Human Generation

This section presents the methodology for deep garment generation. As [2, 17, 21, 32], data dimensionality and topology is fixed by encoding it as body offsets. In addition, by masking body vertices we represent different garment types and separate them from the body, e.g. in a similar fashion to [21, 22]. To compute ground truth offsets, a body-to-garment matching is needed. A dedicated algorithm for this task should be able to correctly register skirt-like garments which have a different topology than the body. In sec. 4.1 we explain details of our data pre-processing. Our proposed model is a Graph Conditional Variational Auto-Encoder (GCVAE). By conditioning on available metadata (pose, shape and tightness), we learn a latent space encoding specific information about garment type and its dynamics (details are given in sec. 4.3). Fig. 5 illustrates the proposed model.

4.1 Data Pre-processing

In order to match among garment and body, we apply non-rigid ICP [4]. Registration is performed once per sequence in rest pose. Due to SMPL low vertex

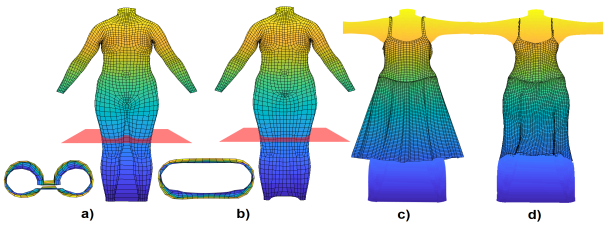

Fig. 4: Dual topology and registration. a) New additional proposed topology, where inner legs are connected. This topology is used for graph convolutions as well. b) Result of Laplacian smoothing of inner leg vertices. It is used only for skirt/dress registration. We show top view of meshes around an imaginary red cutting plane. c) Garment in rest pose. d) Garment registered to body model.

resolution, garment details could be lost. For this reason we subdivide the mesh (and corresponding SMPL model parameters). Head, hands and feet are not used to find correspondences and removing them halves input dimensionality. This yields a final mesh with $N = 14475$ vertices. Finally, note that skirt-like garments do not follow the same topology as SMPL mesh. For this task we introduce a novel topology explained on the subsection below. An example of the registration is shown in Fig. 4. Finally, body to cloth correspondences and garment mask are extracted by nearest neighbor matching.

4.2 SMPL-Skirt Topology

From SMPL body mesh, a 'column' of inner faces of each leg is removed and a new set of faces is created by connecting vertices from both legs, see Fig.4a. New faces are highly stretched, producing noisy garment registrations if used as is, NR-ICP yields optimal results for homogeneous meshes (in garment domain). Because of this, we apply an iterative Laplacian smoothing to vertices belonging to the inner parts of each leg, see Fig.4b for the result. This process is repeated before registration with the corresponding shape of the subject in the sequence in T-pose. This gives a matching between garment and body vertices to compute offsets. For encoding garments as offsets we use body mesh without smoothing, as this process will misbehave for posed bodies. Finally, for graph convolutions, we use the Laplacian matrix corresponding to this new topology for garments of type Dress and Skirt. This ensures that vertex deep features are aggregated with the correct neighbourhood. Afterwards, we transfer body topology to the predicted garment, and it is therefore crucial to use the correct topology for each garment type.

4.3 Network

As shown in Fig. 5, our network is based on a VAE generative model. The goal is to learn a meaningful latent space associated to the garments of any type, shape

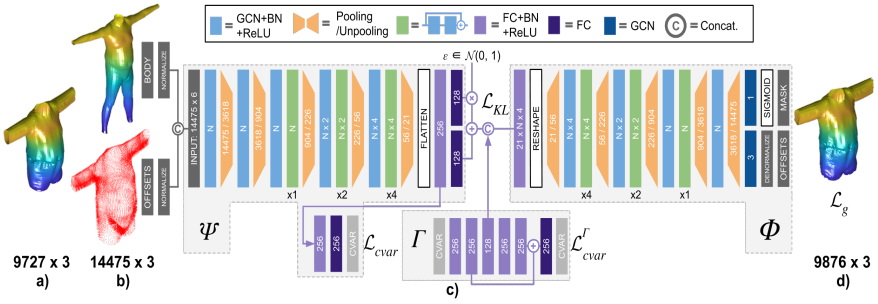

Fig. 5: Model pipeline. a) Input garment b) body and offsets w.r.t. body (Sec. 4.1). Model input is the concatenation of body and offsets. c) Network architecture. Conditional variables (CVAR) are processed by an AutoEncoder. To improve latent space factorization, CVAR are also regressed from the first encoder FC layer. Decoder outputs are offsets and mask. d) Reconstruction of the garment by adding offsets to body and removing body vertices according to mask. We set N as 128.

or with wrinkles which is used to generate realistic draped garments. Garment type and shape are associated to the static state of the garment while wrinkles belong to the dynamics of the garments. Here, we disentangle the latent space between statics and dynamics of the garments, and refer to learnt latent codes as garment code ($z_s \in \mathbb{R}^{128}$) and wrinkle code ($z_d \in \mathbb{R}^{128}$), respectively. To do so, we build two separate networks, one trained on static garments (so called SVAE) and one trained on dynamic garments (so called DVAE). To factorize the latent space from irrelevant parameters to the garment type and shape, we condition SVAE on body shape ($\beta \in \mathbb{R}^{11}$)[4] and garment tightness ($\gamma \in \mathbb{R}^2$). Likewise, DVAE is conditioned on β, γ, body pose ($\theta \in \mathbb{R}^{f\times 72}$) and z_s, where f is the number of frames in a temporal sequence. Let $cvar_s$ and $cvar_d$ be the stacking of conditioning variables of SVAE and DVAE in a single vector. It is worth noting that θ is constant in SVAE so that we do not include it in $cvar_s$. We implement graph convolutions as in [6, 7, 17, 19, 31, 34]. We also include skip connections throughout the whole network.

Architecture. Let $X_s \in \mathbb{R}^{V_T\times 3}$ and $X_d \in \mathbb{R}^{V_T\times 3}$ be offsets computed on static and dynamic samples, respectively. From now on we use subscript s and d for static and dynamic variables and discard them for general cases. SVAE and DVAE have a similar structure with three main modules: encoder $\{cvar^z, z\} = \Psi(\bar{X}, \bar{T})$, conditioning $\{cvar, cvar^z\} = \Gamma(cvar)$ and decoder $\{\bar{X}, M\} = \Phi(z, cvar^z)$, where $M \in \mathbb{R}^{V_T\times 1}$ is the garment mask. Conditioning network Γ is an autoencoder with one skip connection and $cvar^z$ is its middle layer features. The goal of this network is to provide a trade-off between $cvar$ and z. The architecture details are shown in Fig. 5. Note that all GCN layer features (except first and last

[4] We include gender as an additional dimension to the shape parameters.

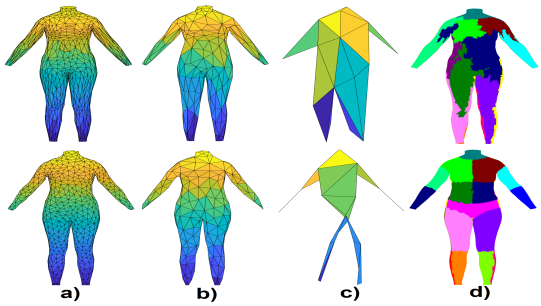

Fig. 6: Mesh hierarchy for pooling. Upper: default [9]. Lower: proposed. a), b) and c) depict the mesh hierarchy used for graph pooling through the model. Observe the difference on spatial distribution at a) and b). c) shows how lowest pooling is more meaningful regarding the segments (one vertex per segment). d) is the visualization of correspondences (receptive field) between highest and lowest hierarchy levels. The proposed pooling yields more meaningful pooling receptive fields w.r.t. body parts.

layers) are doubled in DVAE vs. SVAE. We refer the reader to the supplementary material for more details on the network architecture.

Pooling. We resort to a mesh simplification algorithm [9] to create a hierarchy of meshes with decreasing detail in order to implement the pooling operator. We follow [34] to have vertices uniformly distributed in the graph coarsening. However, this approach does not guarantee a uniform or meaningful receptive field on a high resolution mesh. To achieve a homogeneous distribution of correspondences throughout the body between pooling layers, we define a segmentation (Fig. 6(d)) and forbid the algorithm from contracting edges connecting vertices of different segments. Segmentation contains 21 segments and it is designed such that regions of the body with highest offset variability have smaller segments. Thus, more capacity of the network is available to model those parts. See Fig. 6. Our mesh hierarchy is formed by 6 different levels. The dimensionality of those meshes is: $14475 \rightarrow 3618 \rightarrow 904 \rightarrow 226 \rightarrow 56 \rightarrow 21$, leaving a single node for each segment on the last pooling layer. We use max-pooling in the proposed hierarchy. For unpooling, features are copied to all corresponding vertices of the immediate higher mesh.

Loss. We train conditioning network Γ independently using $L1$ loss and freeze its weights while training VAE. S/DVAE loss is a combination of a garment related term, a *cvar* term and KL-divergence:

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_{cvar} + \lambda_{KL} D_{KL}(q(z|X, cvar)||p(z|cvar)), \tag{2}$$

Garment related term handles offsets, mask (if available), smoothness and collisions:

$$\mathcal{L}_g = \mathcal{L}_o + \lambda_n \mathcal{L}_n + \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c, \tag{3}$$

	Surface	Normals	Mask	KL loss
All	14.3	1.04	0.9518	0.9820
No normals	22.8	1.07	0.9472	0.5966
No mask	92.7	1.19	-	0.8799
No collision	14.7	1.02	0.9522	0.9414
No CVAR	14.8	1.02	0.9520	1.1009
Default pooling	14.9	1.03	0.9390	0.7623

(a)

	Surface	Normals	Mask	KL loss
Top	11.9	1.20	0.9035	0.9536
T-shirt	15.5	1.21	0.9565	1.1701
Trousers	10.9	0.84	0.9475	0.9008
Skirt	21.4	0.79	0.9520	1.0255
Jumpsuit	13.3	1.07	0.9637	0.8788
Dress	16.7	1.06	0.9662	0.9995

(b)

(c) Table 2: (a) Ablation results on the static dataset for all clothes. (b) Ablation results (full model) on the static dataset for each cloth category. Surface and normal errors are shown in mm and radians, respectively.

where $\mathcal{L}_o$ is an L1-norm applied to output offsets. $\mathcal{L}_n$ is the smoothness term based on L1-norm on normals. We found that regular Laplacian loss ensures smoothness at the cost of losing high frequency geometric details, while a normal loss makes output geometry consistent w.r.t. the input. $\mathcal{L}_m$ consists on L1-norm on mask. Finally, $\mathcal{L}_c$ is the collision loss. Given that garments are represented as offsets, we design this loss as:

$$\mathcal{L}_c = max(0, -o \cdot V_N), \tag{4}$$

where o are the output offsets and V_N are the body normals at the corresponding vertices, this penalizes offsets that go within the body. $\mathcal{L}_{cvar}$ is $L1$ loss on encoder $cvar^z$ regressor.

5 Experiments

First, we detail the metrics chosen to analyze the results.

Surface. Given that input and prediction have the same dimensionality and order, we use standard euclidean norm (in mm.).

Normals. Measure of surface quality. We compute normals error based on mesh face normals by their angle difference (in radians) to ground truth normals.

Mask. Garment mask is evaluated by the intersection over union (IoU).

KL Loss. We use KL loss as a measure of quality of latent code factorization and meaningfulness of the latent space.

5.1 Ablation Study

We trained SVAE on an additional dataset of static samples (in rest pose) with 30K samples. 20% of the data is kept for evaluation and the rest for training. The results are shown in Tab. 3a and 3b.

Normals. Looking at the second row of Tab.3a we observe that enforcing a reconstruction consistent with normals significantly reduces surface error and, as expected normals error. However, including normals has a negative impact on KL loss comparing to first row.

# frames	Top	T-shirt	Trousers	Skirt	Jumpsuit	Dress	Avg.
1	21.8/1.24	28.8/1.29	20.7/0.89	37.6/0.92	28.2/1.15	35.5/1.13	29.0/1.10
4	20.1/1.23	28.0/1.28	18.5/0.86	33.2/0.89	26.1/1.09	32.2/1.11	26.1/1.08

Table 4: Ablation results (full model) on the dynamic dataset conditioning on different number of frames. Left: surface error (mm) / Right: normals error (radians).

Mask. As seen in third row of Tab. 3a, both, surface and normals error are significantly higher without mask prediction (comparing to first row).

Collision. Fourth row of Tab. 3a shows how collision loss helps to improve vertex location by pushing collided vertices to their correct position. On the other hand, it is observable a non-significant increase on other losses.

CVARs. As explained in Sec.4.3, conditional variables are regressed from the first FC layer of the encoder to improve latent space factorization. On fifth row of Tab. 3a we can see that, while surface or normals error have no significant differences, KL loss improves.

Pooling. On Sec.4.3 we discussed different approaches for tackling the pooling on a graph neural network. To do this, we built a mesh hierarchy. We compared default mesh simplification algorithm versus our proposed modification. Results are shown in the last row of Tab. 3a. While improvement on surface and normals errors is marginal, this new pooling benefits mask prediction.

Per Garment Category Error. Results per garment are shown in Tab. 3b. Skirts present the highest surface error, as its vertices are further away from the body compared to other garments. Following this reasoning, we find trousers having the less surface error. If we look at normals error, we find an opposite behaviour for skirts, as their geometry is the simplest one. On the other hand we see that upper-body garments present more complex geometries, and therefore, higher normals error. Looking at mask error, we see that garments that cover most of the body have the lowest error. This is due to IoU metric nature, the lower the number of points, the more impact shall have each wrong prediction. Finally, looking at KL loss, we observe the model has difficulties to obtain meaningful spaces for T-shirts. As explained on Sec.3.2, T-shirts category includes open shirts as well, which highly increases class variability. We also see that trousers and jumpsuits have the lowest KL loss.

Learned Latent Space. In Fig. 7a, we show distribution of 5K random static samples computed by t-SNE algorithm. As one can see, the proposed GC-VAE network can group garments in a meaningful space. Interestingly, dress and jumpsuit that share more vertices also share the same latent space. Additionally, we show garment transitions in this space in Fig. 7b. One can see how garments transit between two different topologies (3rd row) or among different genders and shapes (4th row).

We study DVAE model in Tab. 4. We condition DVAE on pose for a single frame vs. four frames. Four frames are selected every 3 frames, resulting in a 12-frame clip. Training the model on a sequence of frames leads to better results

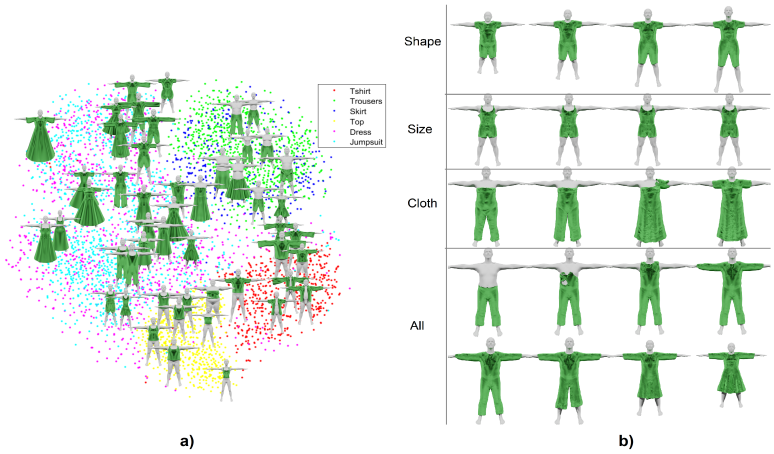

Fig. 7: a) Visualization of the learned latent space for static samples using t-SNE algorithm. b) Transitions of static samples. First three rows: conditioning on shape, tightness or cloth while the rest are fixed. Last two rows: transition of all variables. Variables are linearly graduated.

Fig. 8: Garment reconstruction for sequences. Note that the model has not been trained to keep temporal consistency.

in all garment categories (3mm improvement in average). This is while we do not include any temporal information in the encoder nor any specific sequence prediction loss. DVAE qualitative results for single frames and sequences are shown in Fig.1(right) and Fig.8, respectively.

6 Conclusions

We presented CLOTH3D, the first large scale synthetic dataset of 3D clothed humans. It has a large data variability in terms of body shape and pose, garment type, topology, shape, tightness and fabric. Generated garments also show complex dynamics, providing with a challenging corpus for 3D garment generation. We developed a baseline method using a graph convolutional network trained as a variational autoencoder, and proposed a new pooling grid. Evaluation of the proposed GCVAE on CLOTH3D showed realistic garment generation.

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

References

1. Carnegie-Mellon Mocap Database. http://http://mocap.cs.cmu.edu/
2. Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Detailed human avatars from monocular video. In: 2018 International Conference on 3D Vision (3DV). pp. 98–109. IEEE (2018)
3. Alldieck, T., Pons-Moll, G., Theobalt, C., Magnor, M.: Tex2shape: Detailed full human body geometry from a single image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2293–2303 (2019)
4. Amberg, B., Romdhani, S., Vetter, T.: Optimal step nonrigid icp algorithms for surface registration. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
5. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-garment net: Learning to dress 3d people from images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5420–5430 (2019)
6. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
7. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
8. Dong, Q., Gong, S., Zhu, X.: Multi-task curriculum transfer deep learning of clothing attributes. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 520–529. IEEE (2017)
9. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. pp. 209–216. ACM Press/Addison-Wesley Publishing Co. (1997)
10. Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., Black, M.J.: Drape: Dressing any person. ACM Trans. Graph. **31**(4), 35–1 (2012)
11. Gundogdu, E., Constantin, V., Seifoddini, A., Dang, M., Salzmann, M., Fua, P.: Garnet: A two-stream network for fast and accurate 3d cloth draping. In: IEEE International Conference on Computer Vision (ICCV). IEEE (oct 2019)
12. Lahner, Z., Cremers, D., Tung, T.: Deepwrinkles: Accurate and realistic clothing modeling. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 667–684 (2018)
13. Lin, K., Yang, H.F., Liu, K.H., Hsiao, J.H., Chen, C.S.: Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 499–502. ACM (2015)
14. Loper, M., Mahmood, N., Black, M.J.: Mosh: Motion and shape capture from sparse markers. ACM Transactions on Graphics (TOG) **33**(6), 220 (2014)
15. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
16. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 248 (2015)
17. Ma, Q., Tang, S., Pujades, S., Pons-Moll, G., Ranjan, A., Black, M.J.: Dressing 3d humans using a conditional mesh-vae-gan. arXiv preprint arXiv:1907.13615 (2019)
18. von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: European Conference on Computer Vision (ECCV) (sep 2018)

19. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International conference on machine learning. pp. 2014–2023 (2016)
20. Nikolenko, S.I.: Synthetic data for deep learning. ArXiv **abs/1909.11512** (2019)
21. Patel, C., Liao, Z., Pons-Moll, G.: Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7365–7375 (2020)
22. Pons-Moll, G., Pujades, S., Hu, S., Black, M.J.: Clothcap: Seamless 4d clothing capture and retargeting. ACM Transactions on Graphics (TOG) **36**(4), 73 (2017)
23. Pumarola, A., Goswami, V., Vicente, F., De la Torre, F., Moreno-Noguer, F.: Unsupervised image-to-video clothing transfer. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
24. Pumarola, A., Sanchez-Riera, J., Choi, G., Sanfeliu, A., Moreno-Noguer, F.: 3dpeople: Modeling the geometry of dressed humans. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2242–2251 (2019)
25. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
26. Santesteban, I., Otaduy, M.A., Casas, D.: Learning-based animation of clothing for virtual try-on. In: Computer Graphics Forum. vol. 38, pp. 355–366. Wiley Online Library (2019)
27. Shin, D., Chen, Y.: Deep garment image matting for a virtual try-on system. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (Oct 2019)
28. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 109–117 (2017)
29. Wang, T.Y., Ceylan, D., Popovic, J., Mitra, N.J.: Learning a shared shape space for multimodal garment design. arXiv preprint arXiv:1806.11335 (2018)
30. Wang, T.Y., Shao, T., Fu, K., Mitra, N.J.: Learning an intrinsic garment space for interactive authoring of garment animation. ACM Transactions on Graphics (TOG) **38**(6), 1–12 (2019)
31. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596 (2019)
32. Yang, J., Franco, J.S., Hétroy-Wheeler, F., Wuhrer, S.: Analyzing clothing layer deformation statistics of 3d human motions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 237–253 (2018)
33. Yu, T., Zheng, Z., Zhong, Y., Zhao, J., Dai, Q., Pons-Moll, G., Liu, Y.: Simulcap: Single-view human performance capture with cloth simulation. arXiv preprint arXiv:1903.06323 (2019)
34. Yuan, Y.J., Lai, Y.K., Yang, J., Fu, H., Gao, L.: Mesh variational autoencoders with edge contraction pooling. arXiv preprint arXiv:1908.02507 (2019)
35. Zhang, C., Pujades, S., Black, M.J., Pons-Moll, G.: Detailed, accurate, human shape estimation from clothed 3d scan sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4191–4200 (2017)