GenSim: Unsupervised Generic Garment Simulator

Lokender Tiwari Brojeshwar Bhowmick Sanjana Sinha
TCS Research, India
{lokender.tiwari, b.bhowmick, sanjana.sinha}@tcs.com

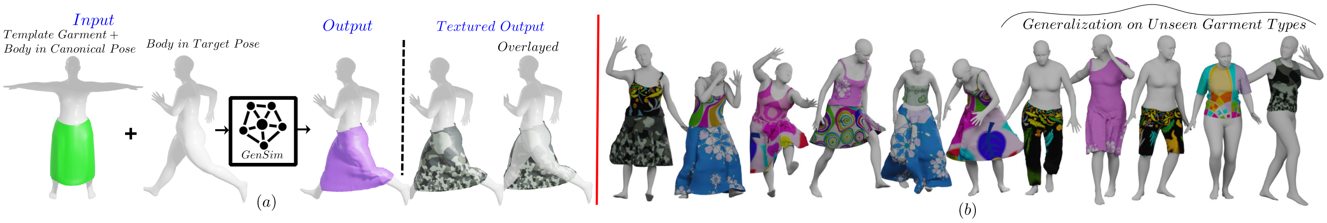

Figure 1. (a). $GenSim$ takes a template garment on the canonical body pose and the target body pose as an input, and deforms the garment to simulate it on the target pose. We can observe in the overlayed output that the simulated garment obeys the geometry of the underlying body shape and the pose (b). $GenSim$ is a **generic** garment simulator. In contrast to other existing methods, our single-trained model of $GenSim$ works for a variety of garments. $GenSim$ trained on tops and skirts from CLOTH3D [4] dataset but generalizes well on unseen garment types such as pants, shorts, tshirt, and tank from VTO dataset [21, 23] and garments from CLOTH3D [4] test set with a variety of body poses and shapes. Please refer to Sec. 4 and Supplementary for more results.

Abstract

In this paper, we propose a novel generic garment simulator to drape a template 3D garment of arbitrary type, size, and topology onto an arbitrary 3D body shape and pose. Existing learning-based methods for 3D garment simulation methods train a single model for each garment type, with a fixed topology. Most of them use supervised learning, which requires huge training data that is expensive to acquire. Our method circumvents the above-mentioned limitations by proposing GenSim, a generic unsupervised method for garment simulation, that can generalize to garments of different sizes, topologies, body shapes, and poses, using a single trained model. Our proposed GenSim consists of (1) a novel body-motion-aware as-rigid-as-possible (ARAP) garment deformation module that initially deforms the template garment considering the underlying body as an obstacle and (2) a Physics Enforcing Network (PEN) that adds the corrections to the ARAP deformed garment to make it physically plausible. PEN uses multiple types of garments of arbitrary topology for training using physics-aware unsupervised losses. Experimental results show that our method significantly outperforms the existing state-of-the-art methods on the challenging CLOTH3D [4] dataset and the VTO [23] dataset. Unlike the unsupervised method PBNS [5], GenSim generalizes well on unseen garments with varying shapes, sizes, types, and topologies draped on different body shapes and poses.

1. Introduction

Simulating 3D garments of varying types and sizes on 3D humans of arbitrary body shapes and poses is an important problem with direct applications in virtual try-on, garment authoring, garment catalog generation, etc. Although Physics-Based Simulation (PBS) approaches are preferable for accurate and realistic simulation of garments on 3D humans, they are computationally expensive and require expert intervention.

Recently, there has been a growing interest in data-driven learning-based simulation approaches [5, 6, 9, 12, 17, 18, 22, 23, 26–29, 32] as they reduce dependency on experts, and the time complexity of the simulation significantly. However, they have several limitations. Many of these methods 1) *Work on fixed garment type:* for instance PBNS [5], SNUG [22], [21], [23], VBones [17] and DD3DG [32] can simulate **only** the garment template/type which is used for training thereby requiring a new model for each garment type. 2) *Work on fixed body shapes*: VBones [17] and PBNS [5] work on the fixed body shape used for training. 3) *Assume fixed garment topology*: methods such as DeepDraper [27], and TailorNet [18] assume different sizes of garments to have the same number of vertices, hence cannot model loose and long garments such as skirts and dresses, which require more number of vertices to accurately represent the geometry (wrinkles and folds). A comparison of various methods based on their ability to handle various

Table 1. Comparison of state-of-the-art garment simulation methods based on the generalization ability of a single trained model. At test time, our method can handle garment and body property variations using a *single trained model* that is generalized for all garment and body types. *DD3DG cannot handle poses that are very different from the training motion types.

		variations during testing				
		body shape	body pose	garment Types	garment topologies	body topologies
Supervised	DD3DG [32]	✓	✓*			
	NCLOTH [8]	✓	✓			
	VBones [17]		✓			
	TAILORNET [18]	✓	✓			
	LVTON [21]	✓	✓			
	GARNET [10]	✓	✓			
	DeepDraper [27]	✓	✓			
	DeePSD [6]	✓	✓	✓	✓	
	GarSim [28]	✓	✓	✓	✓	
Self-Supervised	SNUG [22]	✓	✓			
Un-Supervised	PBNS [5]		✓			
	GenSim (ours)	✓	✓	✓	✓	✓

variations during testing is shown in Table. 1.

The existing supervised methods are trained primarily by minimizing $L2$ distance w.r.t the PBS data. Minimizing such $L2$ distance jointly for multiple tight and loose garments tends to learn average displacement and compromise the physical plausibility of the deformation.

PBNS [5] is an unsupervised garment deformation model. Nevertheless, it is a pose-space deformation [13] model that takes only the body pose parameters as input and predicts a garment-specific pose deformation matrix. Hence it cannot be trained for multiple garments simultaneously. In this paper, we aim to circumvent the limitations of PBNS by proposing a garment simulator that takes both the garment and body as input, along with other factors affecting garment deformation on a 3D human body. In particular, we propose a two-stage method to deform a garment over a 3D human body (refer Fig. 2). First, given a template body pose and the target pose (therefore, the relative motion between these poses), we estimate body motion-aware as-rigid-as-possible [24] garment deformation iteratively by avoiding garment-body collisions. This stage roughly aligns the template garment on top of the target body but doesn't ensure physical properties such as local smoothness, or the forces acting on the clothes due to gravity, stretching, bending, etc. In the next stage, the ARAP-deformed garment, the body geometry, and the motion information from the template pose are sent to a Physics Enforcing Network (PEN) to ensure the physical plausibility of the garment by adding a correction to the ARAP-deformed garment. The PEN creates an encoded garment graph with latent node and edge features representing the body and garment features. The latent features get processed by a message-passing graph network and decoded to generate a physically plausible garment deformation. The message-passing graph network representation helps our method generalize to an arbitrary number of vertices/topologies of both the garment and body meshes. Our motion-aware garment deformation algorithm ensures that there are no body-to-garment collisions. PEN leverages this information to ensure that the predicted garment displacement locally avoids the body-garment collision.
Our contributions can be summarized as follows:

- We propose $GenSim$, a generic method for unsupervised 3D garment simulation. To the best of our knowledge, our method is the first *unsupervised* garment simulator that can be trained *simultaneously* for *multiple types* of garments of varying sizes, and topology, and bodies of different shapes and poses. It generalizes on unseen garments of different types, and sizes, along with different body shapes and poses.
- We propose a novel *Motion-Aware ARAP Garment Deformation* method that roughly aligns the template garment on the target body pose.
- We propose a *Physics Enforcing Network (PEN)* that corrects the output of the Motion-Aware ARAP module to make the deformation physically plausible. PEN is trained in an unsupervised manner with the help of loss functions for physical plausibility.

2. Related Work

In this section, we review learning-based methods for garment deformation in three categories: *supervised*, *self-supervised*, and *unsupervised*. A comparison table with all these categories is shown in Table. 1.

Supervised methods mainly rely on the $L2$ loss between the predicted garment deformation and the ground-truth PBS data for training. Majority of the existing methods [4, 6, 8, 10, 14, 17, 18, 26, 27, 30] follow the supervised approach. DeepDraper [27] and TailorNet [18] train different models for different types and topologies of garments. TailorNet [18] learns low-frequency and garment style-specific high-frequency models and combines them using RBF kernel to finally get a body shape, pose, and style-aware garment deformation model. Similarly, DeepDraper [27] uses coupled geometric and multiview-perceptual constraints to train for each type of garment. DeepWrinkles [12] proposes a learned Pose Space Deformation [13] conditioned on temporal features. LVTON [21] learns a garment-specific non-linear mapping for pose-aware garment deformation. Methods like DeePSD [6] and CLOTH3D [4] need a huge amount of PBS data to train their respective models. Moreover, as mentioned by the respective authors in their papers, these methods fail to work on loose garments such as dresses, long skirts, etc. In a very recent attempt, VBones [17] propose a method to predict deformation for loose garments. It uses the concept of virtual bones and concepts of skinning with these bones to animate the garment based on the underlying body sequence. Another recent method,

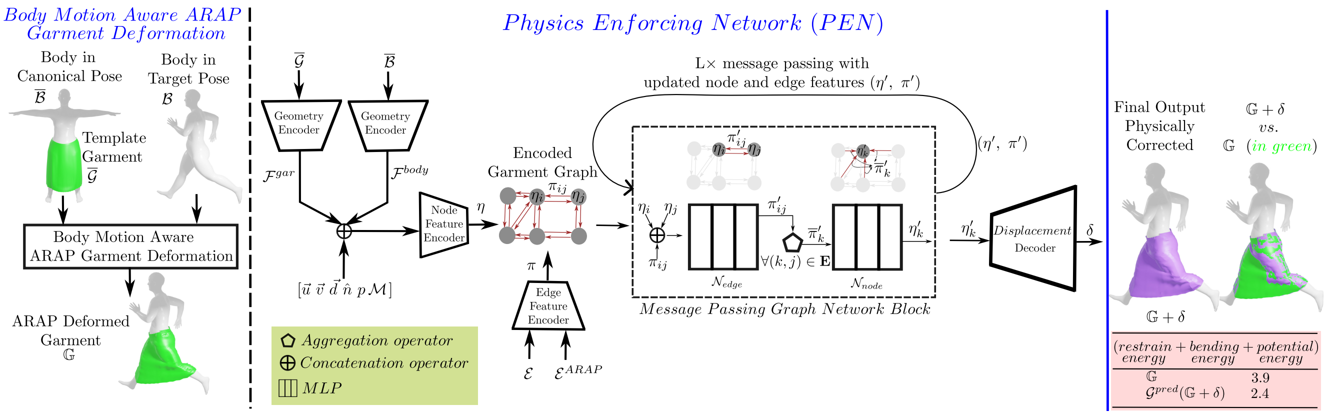

Figure 2. *GenSim* has a two-step approach for garment simulation. The first step is a novel Motion-Aware ARAP garment deformation, which takes as input the template garment draped on a body in the canonical pose, and roughly aligns the template garment on the target body pose (Sec. 3.1). The second step is a Physics Enforcing Network (PEN) which takes the ARAP deformed garment $\mathbb{G}$ as input and predicts the per-vertex displacements δ to make $\mathbb{G}$ physically plausible. PEN first creates an encoded garment graph with latent node η and edge π features (Sec. 3.2). The latent features are obtained by encoding various factors such as the geometry of both the garments and the body ($\mathcal{F}^{gar}$, $\mathcal{F}^{body)}$), relative body motion $\vec{d}$, relative position of garment vertices ($\vec{u}$, $\vec{v}$), vertex masses $\mathcal{M}$, etc, (Sec. 3.2.1). The encoded graph is then processed through a message-passing graph network block (Sec. 3.2.2). The final processed graph node features are then used by the decoder to predict per vertex displacements δ which are added to $\mathbb{G}$ to get the final physically corrected plausible garment (Sec. 3.2.3). The overall energy of $\mathbb{G} + \delta$ is low compared to $\mathbb{G}$ which shows PEN is making ARAP deformed garment $\mathbb{G}$ physically plausible.

DD3DG [32] learns a per-garment generative model for motion-guided garment deformation. However, both [32] and [17] are garment specific and cannot handle variable topology garments. For every new type of loose garment, both needs to train a new model.

Self-supervised methods such as SNUG [22], use the previously inferred frames to drive the dynamics of garment vertices and predict the deformation of the same on the target body pose in a self-supervised manner. This method is also a *garment specific model*, i.e, it can only be evaluated on the same template garment which is used for training the model.

Unsupervised methods such as PBNS [5] do not use ground-truth data, instead, they cast garment deformation as an energy minimization problem, and derive losses based on various energies such as potential energy due to gravity, strain energy, bending energy, etc. PBNS trains a single model for each *outfit and body pair* which is a major limitation. Practically, it is infeasible to train for every outfit and body pair. Our method *GenSim* alleviates this problem. *GenSim* being an unsupervised method, it can be trained simultaneously for multiple variations of garment type, size, shape, body shapes, and poses. *GenSim* is capable of generalization on unseen garment shape sizes, topologies, and body shapes and poses using a single trained model.

3. Methodology

GenSim simulate the template garment $\overline{\mathcal{G}}$ on the target body $\mathcal{B}$ in an arbitrary pose in two-steps. In the first step, it roughly aligns the template garment on the target body pose using our novel Body Motion-aware as-rigid-as-possible garment deformation module (Sec.3.1). The first step doesn't account for the physical constraints. Hence, we introduce a novel Physics Enforcing Network (PEN) (Sec.3.2), that adds the corrections to the roughly aligned garment in the second step to make it physically plausible. Next, we describe both steps in detail.

3.1. Step 1 - Body Motion-Aware ARAP

The traditional ARAP algorithm [24] has been widely used to edit the meshes while keeping the local rigidity of the mesh as much as possible. It takes an initial mesh (vertices and faces), a set of handle vertices, and their displaced locations. Handle vertices are the set of mesh vertices usually manually selected and displaced by a user. ARAP keeps the position of handle vertices fixed to their displaced locations, and estimates the displacements for the non-handle vertices preserving the local rigidity in the mesh as much as possible. This traditional ARAP cannot be applied in our scenario because, **1.** *the set of handle vertices are unknown and can be different for every target body pose*, **2.** *the articulated human body under the garment which will act as an obstacle while deforming non-handle vertices*.

Our proposed approach to deform a garment in the presence of the human body as an obstacle is explained in the Algorithm 1. The handle vertices are the garment vertices under the direct influence of the force exerted by the motion of the underlying body. Let us assume, we have the indexes of handle vertices of the garment in the set $\mathcal{H}$ and their displaced locations in $\tilde{\mathcal{G}}_i$, $\forall i \in \mathcal{H}$. We will discuss in Sec. 3.1.1 how to automatically obtain $\mathcal{H}$ and $\tilde{\mathcal{G}}$. We first compute the displacement vector $\vec{g}_i$ for each of the i^{th} handle vertex with respect to its position in the template garment (Alg. 1 line 2). Instead of setting the handle vertices di-

```
Algorithm 1: Motion Aware ARAP Garment Deformation
Inputs: $\tilde{\mathcal{G}}$, $\overline{\mathcal{G}}$, $\mathcal{B}$, $\hat{n}$, $\mathcal{H}$, $t$,   Output: $\mathbb{G}$
1  $\mathcal{I} = \{1, ..., m\}$   // index set of garment vertices
2  $\vec{g}_i = \tilde{\mathcal{G}}_i - \overline{\mathcal{G}}_i \quad , \forall i \in \mathcal{H}$
3  $\Omega_i = \overline{\mathcal{G}}_i, \quad \forall i \in \mathcal{H}$
4  for $itr \in \{1 \ldots nitr\}$ do
5      $\mathcal{S}_i = \Omega_i + \hat{g}_i \times (||\vec{g}_i||\, t), \quad \forall i \in \mathcal{H}$
6      $\mathbb{G} = \mathcal{ARAP}(\mathcal{S}, \overline{\mathcal{G}})$
7      $\nu_i = FindClosest(\mathbb{G}_i, \mathcal{B}) \quad , \forall i \in \{\mathcal{I} \setminus \mathcal{H}\}$
8      $\zeta_i = (\mathbb{G}_i - \mathcal{B}_{\nu_i}) \cdot \hat{n}_{\nu_i} \quad , \forall i \in \{\mathcal{I} \setminus \mathcal{H}\}$
9      $\varphi = \{i \mid \zeta_i \leq 0, \quad \forall i \in \{\mathcal{I} \setminus \mathcal{H}\}$
10     $\mathbb{G}_i = \mathbb{G}_i + \hat{n}_{\nu_i} |\zeta_i|, \quad \forall i \in \varphi$
11     $\Omega_i = \mathcal{S}_i, \quad \forall i \in \mathcal{H}$
12 end for
```

```
Algorithm 2: Handle Vertices for ARAP
Inputs: $\overline{\mathcal{B}}$, $\mathcal{B}$, $\overline{\mathcal{G}}$, $\epsilon$, $\omega$,   Outputs: $\tilde{\mathcal{G}}$, $\mathcal{H}$
1  $\mathcal{I} = \{1, ..., m\}$   // index set of garment vertices
2  $\nu_i = FindClosest(\overline{\mathcal{G}}_i, \overline{\mathcal{B}}) \quad , \forall i \in \{\mathcal{I} \setminus \omega\}$
3  $\vec{d}_i = \mathcal{B}_{\nu_i} - \overline{\mathcal{B}}_{\nu_i} \quad , \forall i \in \{\mathcal{I} \setminus \omega\}$
4  $\vec{u}_i = \overline{\mathcal{G}}_i - \overline{\mathcal{B}}_{\nu_i} \quad , \forall i \in \{\mathcal{I} \setminus \omega\}$
5  $\tau_i = \hat{u}_i \cdot \hat{d}_i \quad , \forall i \in \{\mathcal{I} \setminus \omega\}$
6  $\mathcal{H} = \{i \ |\tau_i \ \leq \ \epsilon\}, \quad \forall i \in \{\mathcal{I} \setminus \omega\}$
7  $q_i = \frac{\vec{d}_i \cdot \vec{u}_i}{||\vec{d}_i||} \quad , \forall i \in \{\mathcal{I} \setminus \omega\}$
8  $\tilde{\mathcal{G}}_i = \overline{\mathcal{G}}_i + \hat{d}_i\ (||d_i|| - q_i) \quad , \forall i \in \mathcal{H}$
9  $\tilde{\mathcal{G}}_i = \overline{\mathcal{B}}_i + \hat{n}_i\ ||u_i|| \quad , \forall i \in \omega$
10 $\mathcal{H} = \mathcal{H} \cup \omega$
```

rectly to their final location i.e., $\tilde{\mathcal{G}}$, we move the handle vertices **iteratively** towards their final location in the direction of the vector $\vec{g}_i$ and solve for the non-handle vertices $\mathbb{G}$ using ARAP (Alg. 1 line 6). The step size of the movement is controlled by t. Next, we solve for the collision (*if any*) of non-handle vertices with the body. For each deformed non-handle vertex $\mathbb{G}_i$, we find the index ν_i of the closest body vertex ($\mathcal{B}_{\nu_i}$) (Alg. 1 line 7) and its unit normal ($\hat{n}_{\nu_i}$). The projection of the body to the garment vector onto the unit normal vector can indicate the collided vertices (Alg. 1 line 8-9). We resolve the collision by moving the collided garment vertices towards the closest body vertex normal $\hat{n}_{\nu_i}$ with a magnitude of $|\zeta|$ (Alg. 1 line 10). We fix the total number of iterative steps $nitr = 10$ and $t = 1.0/nitr$ for all our experiments. We observe that the movement of the garment handle vertices directly to their final location i.e., $t = 1.0$ and $nitr = 1$, leads to significant collision with the underlying body. In many cases, garment collision would be on the opposite side of the body, which would be difficult to solve. Hence, we fixed the location of handle vertices in a step-wise manner (Alg. 1 line 5) and resolve collision with every ARAP estimate (Alg. 1 lines 7-10).

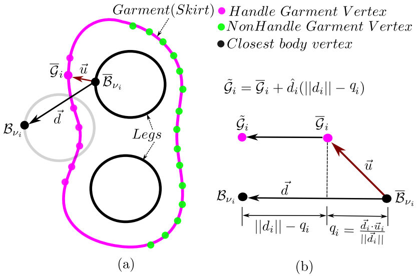
$Garment(Skirt)$
$Handle\ Garment\ Vertex$
$NonHandle\ Garment\ Vertex$
$Closest\ body\ vertex$
$\overline{\mathcal{G}}_i$
$\vec{u}$
$\overline{\mathcal{B}}_{\nu_i}$
$\vec{d}$
$\mathcal{B}_{\nu_i}$
$Legs$
$\tilde{\mathcal{G}}_i = \overline{\mathcal{G}}_i + \hat{d}_i(||d_i|| - q_i)$
$\tilde{\mathcal{G}}_i$
$\overline{\mathcal{G}}_i$
$\vec{u}$
$\mathcal{B}_{\nu_i}$
$\vec{d}$
$\overline{\mathcal{B}}_{\nu_i}$
$||d_i|| - q_i$
$q_i = \frac{\vec{d}_i \cdot \vec{u}_i}{||\vec{d}_i||}$
(a)
(b)

Figure 3. Handle vertices location estimation: The grey circle shows the leg moved due to motion. Refer Sec. 3.1.1 for details.

3.1.1 Handle Vertices for Motion-Aware ARAP

Let $\overline{\mathcal{B}}$ and $\mathcal{B}$ contain the vertices of the body in the canonical and the target body poses respectively, and ω is the index set of the garment pin vertices. We first find the relative motion $\vec{d}_i$ of the body vertex closest to each template garment vertex (Alg. 2 line 2 & 3). Similarly, we compute the relative position vector $\vec{u}_i$ of the template garment vertex with respect to the closest body vertex (Alg. 2 line 4). The angle between $\vec{u}_i$ and $\vec{d}_i$ can indicate the set of garment vertices that are under the direct influence of the force due to the body motion (Alg. 2 line 6). Fig.3(a) shows an intuitive diagram where the vertices in magenta would be the handle vertices, and in greens are the non-handle vertices (assuming only one leg has been moved). The locations of the handle vertices are computed by moving them along the direction of motion with a magnitude $(||d|| - q)$ (Alg. 2 lines 7-8). We move the garment pin vertices such that their relative positions with respect to the body surface pin area remains approximately the same (Alg. 2 lines 9). We finally update the handle vertices index set $\mathcal{H}$ by adding pin vertices to that set so that ARAP doesn't change the location of pin vertices.

3.2. Step 2 - Physics Enforcing Network

The Body Motion-aware ARAP deformed garment $\mathbb{G}$ obtained after step-1 may not be physically plausible as it is deformed under as-rigid-as-possible constraints. To make is physically plausible, we introduce Physics Enforcing Network (PEN) that for every vertex predicts a displacement or correction δ. The corrections are added to the $\mathbb{G}$ to make it physically plausible. Refer to Fig. 2, we first encode the whole garment into a graph, where each node and edge corresponds to a garment mesh vertex and mesh edge. We associate a latent feature, estimated using factors affecting the physics of the garment deformation to each graph node and edge. Our choice of using graph-based representation is to make our network invariant to the node and edge permutations and number of garment vertices and edges, therefore, to enforce arbitrary relational inductive bias via edge connections [3, 14, 15, 19].

3.2.1 Encoded Garment Graph

The physical plausibility of the deformed garment depends on several factors such as the shape and size of the template garment, the body motion, etc. We encode such features into the garment graph node feature. Similarly, we encode the relative position of the edge end-vertices of the garment mesh into the edge features of the garment graph. Let η_i and π_l denote the i^{th} and l^{th} latent node and edge feature vectors of the encoded garment graph respectively.

Latent Node Features: We first encode the geometry of each vertex of the template garment and the underlying body mesh using a geometry encoder. Let $\mathcal{F}^{gar}_i$ and $\mathcal{F}^{body}_{\nu_i}$ denote the latent geometry feature of i^{th} template garment vertex and its closest body vertex $\overline{\mathcal{B}}_i$. The latent geometric feature vectors are concatenated with the relative position of garment vertex w.r.t its closest canonical pose body vertex $\vec{u}_i = \overline{\mathcal{G}}_i - \overline{\mathcal{B}}_{\nu_i}$, the relative position of ARAP deformed garment vertex w.r.t its closest target pose body vertex $\vec{v}_i = \mathbb{G}_i - \mathcal{B}_{\nu_i}$, the normal vector $\hat{n}_i$ of the body vertex $\mathcal{B}_{\nu_i}$, the pin vertex indicator p_i ($p_i = 1$ for pin vertices else 0), the body motion vector $\vec{d}_i$, and the mass $\mathcal{M}_i$ of the garment vertex, which is proportional to the area of its associated faces. The concatenated input $[\mathcal{F}^{gar}, \mathcal{F}^{body}, \vec{u}, \vec{v}, \vec{d}, \hat{n}, p, \mathcal{M}]$ is processed through a node encoder to get the latent node feature vector η_i.

Latent Edge Features: We find the relative position of edge end-vertices in $\overline{\mathcal{G}}$ and $\mathbb{G}$ as $\mathcal{E}_{ij} = \overline{\mathcal{G}}_j - \overline{\mathcal{G}}_i$ and $\mathcal{E}^{ARAP}_{ij} = \mathbb{G}_j - \mathbb{G}_i$ respectively. Encoding both $\mathcal{E}_{ij}$ and $\mathcal{E}^{ARAP}_{ij}$ using latent edge feature π helps the network to correct any significant elongation/compression in the ARAP-deformed garment edges.

3.2.2 Message Passing Graph Network Module

We use a message-passing algorithm similar to [3] where every node aggregates the information/features from its incident edges, process them to update its own feature, and broadcasts to its neighboring nodes. Similarly, every edge takes the features from its end nodes and updates its own features. We repeat the message-passing for L steps and get the final updated encoded garment graph and pass it to the displacement decoder. The node and the edge update function ($\mathcal{N}_{edge}(\cdot)$, $\mathcal{N}_{node}(\cdot)$) are approximated using MLP, and we use the average operation as the aggregation function. Refer, to supplementary material for the detailed algorithm.

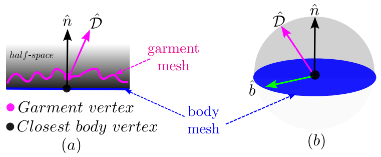

Figure 4. Body-aware direction of displacement to avoid collision with the local body surface. Refer to Sec. 3.2.3 for detail.

3.2.3 Displacement Decoder

The final encoded garment graph node features are used by the decoder to produce per vertex direction $\hat{\mathcal{D}}$ and the magnitude w of the displacement, which is added to the ARAP deformed garment vertices so that the final garment deformation becomes physically plausible. The final predicted deformed garment $\mathcal{G}^{pred} = \mathbb{G} + \hat{\mathcal{D}}\, w$.

To ensure that the direction of displacement $\hat{\mathcal{D}}$ of a garment vertex $\mathbb{G}_i$ is not inducing any collision with the **local** body surface patch around its closest body vertex $\mathcal{B}_{\nu_i}$, we restrict it to be outwards to the local body surface patch i.e., in the half-space towards the unit normal $\hat{n}_i$ of the body vertex $\mathcal{B}_{\nu_i}$. Consider, Fig. 4 where a garment vertex and its closest body vertex are shown in magenta and **black** respectively, and $\hat{n}$ is the body vertex unit normal. The direction $\hat{\mathcal{D}}$ can be in the black gradient region in Fig.4(a), which is the half-space created by the plane (blue) in Fig.4(b) orthogonal to the unit normal $\hat{n}$. The vector $\hat{\mathcal{D}}$ can be obtained as a convex combination of vectors $\hat{n}$ and $\hat{b}$. Let $[b^1, b^2, b^3]$ are the coefficients of the $\vec{b}$, and $\hat{b} = \vec{b}/||\vec{b}||$. Since, $\vec{b}$ is orthogonal to $\hat{n}$, we have $\hat{n} \cdot \vec{b} = 0$. Following this we can write $b^3 = \frac{-b^1n^1 - b^2n^2}{n^3}$, where $[n^1, n^2, n^3]$ are the coefficients of the known body vertex unit normal $\hat{n}$.

The decoder predicts the coefficients b^1, b^2, a convex combination factor ρ, and the magnitude of the displacement w for each garment vertex. The third coefficient b^3 of the direction vector can be estimated from the predicted b^1 and b^2 as explained above. The unit direction vector $\hat{\mathcal{D}}$ is then obtained as $\hat{\mathcal{D}} = \vec{\mathcal{D}}/||\vec{\mathcal{D}}||$, where $\vec{\mathcal{D}}$ can be estimated as shown below in Eq. 1.

$$\vec{\mathcal{D}} = |\rho|\ \hat{n} + (1 - |\rho|) * \frac{\rho}{|\rho|}\ \hat{b} \tag{1}$$

ρ is the output of $Tanh$ activation, hence we use $|\rho|$, to ensure the convex combination ($0 \leq |\rho| \leq 1$) of $\hat{n}$ and $\hat{b}$. To consider both $\hat{b}$ and $-\hat{b}$ as valid solutions(since $\hat{n} \cdot \hat{b} = \hat{n} \cdot$ -$\hat{b} = 0$) we multiply $\hat{b}$ with the factor $\rho/|\rho|$ as this factor can be either $+1$ or -1 and is predicted by the decoder.

3.3. Training Losses

We train Physics Enforcing Network (PEN) end-to-end using unsupervised training losses.

Restrain Energy: The first loss is fabric-aware that resists significant deformation of the garment edges. The length of an edge (i, j) in the template garment is $||\overline{\mathcal{E}}_{ij}||$, and the corresponding edge length in the deformed garment is $||\overline{\mathcal{E}}^{pred}_{ij}||$. The φ_{ij} captures the absolute difference between the length (Eq. 2). The total restrain energy can be computed as

$$\varphi_{ij} = \left|\ ||\mathcal{E}^{pred}_{ij}|| - ||\overline{\mathcal{E}}_{ij}||\ \right| \tag{2}$$

$$\ell_{str} = \sum_{(i,j)\in\mathbf{E}} \left(\frac{\varphi_{ij} - \mu_{fabric}}{\sigma_{fabric}} \right)^2 \tag{3}$$

where, μ_{fabric} and σ_{fabric} are the fabric-specific mean and standard deviation of absolute edge length differences. The μ_{fabric} and σ_{fabric} can be estimated by fitting a Gaussian model to a small subset of physics-based simulation frames of different garment types (shirt, trousers, tops, etc), or simple cloth under different obstacle motions (need not to be the human body). We took a total of random 50 significantly different simulated frames of garments of varying types (randomly chosen shirts, t-shirts, trousers, skirts, and tops). The garments, clothes, body, and obstacle motions used are entirely different from the dataset used in this paper. Refer supplementary for the plot of μ_{fabric} and σ_{fabric}.

Bending Energy: The bending loss term measures the energy due to the angle between two adjacent garment faces [7, 25, 31]. We model it as

$$\ell_{ben} = \sum_{(i,j)} \frac{||\mathcal{E}_{ij}^{pred}||^2}{4(A_1 + A_2)} \left(\frac{\theta^2}{2}\right) \tag{4}$$

where, A_1 and A_2 are the areas of the adjacent faces, $\mathcal{E}_{ij}^{pred}$ is the common edge connecting two adjacent faces, and θ is the dihedral angle between the adjacent faces.

Pin Vertices Loss: We restrict the movement of the pin vertices from their respective location in the ARAP deformed garment $\mathbb{G}$ which prevents the garment to fall down. We apply the $L2$ regularization loss between vertices in $\mathbb{G}$ and $\mathcal{G}^{pred}$. Here, ω is the index set of pin vertices.

$$\ell_{pin} = \sum_{i \in \omega} || \, \mathbb{G}_i - \mathcal{G}_i^{pred} \, ||^2 \tag{5}$$

Potential Energy: The potential energy loss term measures the energy due to the gravitational force. We model it as below, where $\mathbf{g}$ is the gravitational acceleration, z_i is the height of the final deformed garment vertex $\mathcal{G}_i^{pred}$, and $\mathcal{M}_i$ is the mass of the garment vertex. We compute potential energy loss for all the non-pin vertices.

$$\ell_{pe} = \sum_{i \in \{\mathcal{I} \setminus \omega\}} \mathcal{M}_i \, \mathbf{g} \, z_i \tag{6}$$

Mesh Smoothness Loss: To enforce locally smooth surfaces of predicted garment mesh $\mathcal{G}^{pred}$, we apply Laplacian smoothing loss with cotangent weights. Here, $\triangle$ is the Laplacian smoothing operator.

$$\ell_{sm} = \triangle(\mathcal{G}^{pred}) \tag{7}$$

Garment-to-Body Collision Penalty Loss: Our direction of displacement by design avoids collisions locally (Sec. 3.2.3). However, if the garment vertex needs to move a lot i.e., when the magnitude of the displacement is large, in such cases even though the direction is in the right half-space (Sec. 4) the vertex may collide with the non-local body surfaces. This happens mainly when the garment lies between the two body parts, where the normal of both the body parts are facing each other (e.g., the garment lies between the folded leg i.e., between the back side of the thigh and the calf muscle body area). To handle those cases we apply a garment-to-body collision penalty loss as below:

$$\ell_{col} = \sum_{i \in \mathcal{I}} \max(-\hat{n}_i(\mathcal{B}_{\nu_i} - \mathcal{G}_i^{pred}), \; \epsilon_{coll}) \tag{8}$$

where, $\mathcal{B}_{\nu_i}$ is the body vertex closest to the final deformed garment vertex $\mathcal{G}_i^{pred}$, and $\hat{n}_i$ is the unit vertex normal of $\mathcal{B}_{\nu_i}$. ϵ_{coll} is the collision threshold for robustness. Please see the supplementary for an example. PEN is trained using the total loss $\mathcal{L}$ (Eq. 9). An ablation on the loss term is shown in the Sec. 4.4.

$$\mathcal{L} = \gamma_1 \ell_{str} + \gamma_2 \ell_{ben} + \gamma_3 \ell_{pin} + \gamma_4 \ell_{pe} + \gamma_5 \ell_{sm} + \gamma_6 \ell_{col} \tag{9}$$

Here, $\gamma' s$ are the balancing weights of the loss terms.

4. Experiments

We train *GenSim* on a subset of 175 sequences with tops and skirts outfits from the training split of the CLOTH3D [4] dataset and we test on 60 sequences from the test split from this dataset.

4.1. Results on CLOTH 3D Test Set

Quantitative: To demonstrate that our unsupervised method predicts more accurate physically plausible garment deformations, we train two variants of *GenSim* - one with our main unsupervised losses (Eq. 9) as described in Sec. 3.2, and another using $L2$ loss on predicted garment vertices. We compare both variants on the test set using metrics for average edge compression/elongation w.r.t their length in the template garment, garment surface quality smoothness measured using the Laplacian loss term as in Eq. 7, and the percentage of garment vertices collisions with the body. As the per results in Table 2, *GenSim* trained with physics-based losses produces better quality garment deformations owing to lower edge distortion, smoother garment surfaces, and significantly lower body-to-garment collision. In the first variant, the $L2$ loss learns the ground-truth data in an average sense, which leads to elongation/compression of some edges and collisions. On the other hand, unsupervised losses in *GenSim* compute the garment deformation that reduces the overall energy of the garment, while restraining edge lengths from changing too much, and also avoiding collisions, thereby, producing better results.

Table 2. Quantitative Evaluation on CLOTH3D Test Set

Training Strategy	Edge (mm)	Smoothness	Collision (%)
Supervised L2 + Smoothness + Collision Loss	3.8	0.0013	10.1
Unsupervised Physics Losses	**1.9**	**0.0008**	**0.6**

Qualitative: Sample qualitative results of *GenSim* on various unseen body shapes, poses, and garments of various types, sizes, and topologies are shown in Fig. 5. *GenSim* produces physically plausible simulations.

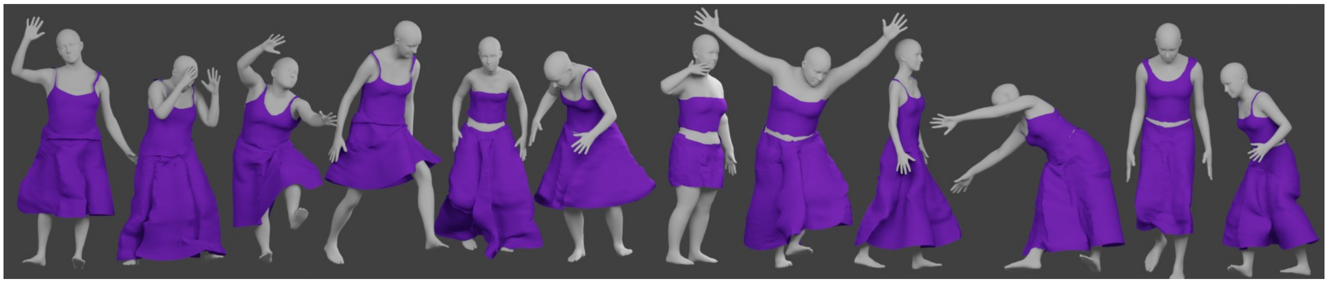

Figure 5. **Sample results on Test Set:** *GenSim* produces physically plausible deformations for varying body shapes and poses, and varying garment types and topologies. Check textured results of some of these outputs in Fig. 1(b), and more results in Supplementary.

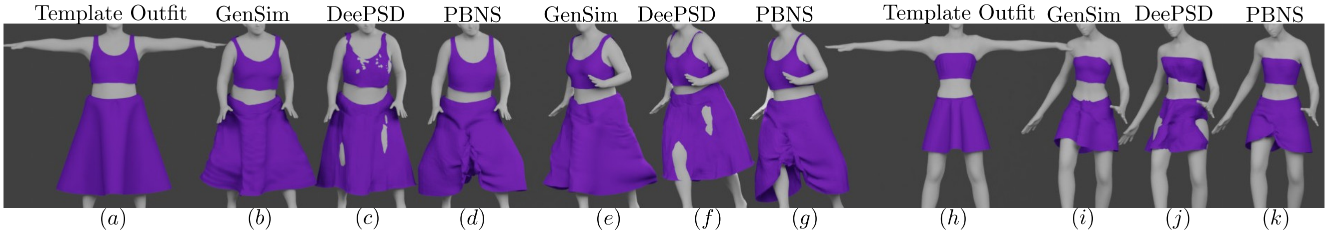

Figure 6. **Comparison with DeePSD and PBNS**. DeePSD produces inferior results on the loose garments while PBNS produces unrealistic artifacts between legs which is quite similar to the reported results in their paper in the case of skirts. PBNS is trained and tested on the same garment template. DeePSD [6] requires **supervised** training using approx $7.5k$ sequences, however, for a fair comparison with ours, we have trained it only on 175 sequences similar to *GenSim* . This shows *GenSim* trained in an **unsupervised** manner using a small training set (175 sequences) accurately learns the physics of deformation, and can produce more accurate physically plausible results than DeePSD.

Comparison with PBNS [5], and DeePSD [6]: We qualitatively compare with PBNS and DeePSD on CLOTH3D test samples. Since PBNS is an outfit-body specific method, we trained two PBNS models one for each outfit-body pair in Fig. 6(a) & 6(h). DeePSD is trained on the same training set of tops and skirts as *GenSim*. **Note:** For PBNS both the training and testing garment template is the same, whereas the testing garment template is unseen for both DeePSD and *GenSim* during training. DeePSD suffers from collisions (in Figs. 6(c), (f) & (j)), PBNS produces similar artifacts for skirts (between legs) as mentioned in their paper [5] (Figs. 6(d), (g) & (k)).

4.2. Generalization on Unseen Garment Types

We evaluate the generalization capability of *GenSim* by evaluating it on unseen garment types (dress, t-shirt, pant, shorts, tank) of different topologies from VTO-dataset [23]. The results shown in Fig. 7 show that despite being trained only on tops and skirts *GenSim* generalized well on unseen garments. We show qualitative comparison with VTO-COLL [23] in Fig.8. Refer, to supplementary material for additional results and comparison.

On random YouTube video: We show results of *GenSim* draping unseen garments (pants, shorts, dress, and tank) on unseen body shapes and poses in a frame of random YouTube video in Fig. 9. We use PARE [11] to estimate the body shape and pose of humans from the image.

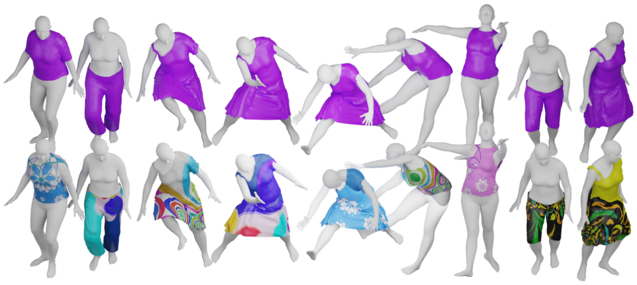

Figure 7. **Generalization** on Unseen Garment Types.

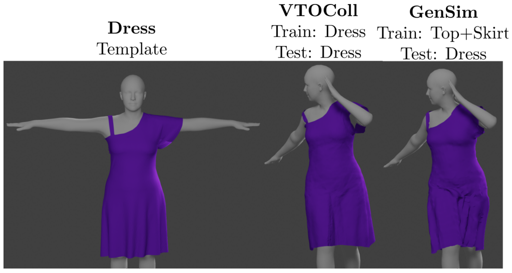

Figure 8. **Comparison with [23]**. VTOCOLL [23] is a **self-supervised** method and *trained and tested on the same dress template*, whereas *GenSim* which is an **unsupervised** method never seen dress during training, generalizes well on unseen garment dress, and produces physically plausible simulations.

4.3. Implementation Details

We have implemented our method using PyTorch [1]. We use PointNet [20] for our geometry encoders, the node and the edge feature encoders are MLPs. We train *GenSim* using adam optimizer for 30 epochs with an ini-

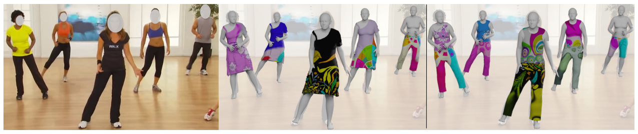

Figure 9. Draping **unseen garment types** (pants, shorts, dress and tank) on **unseen persons** in a random YouTube video frame.

tial learning rate 0.0001 and reduce it by a factor of 10 at 20^{th} epoch. We set $\gamma_1 = \gamma_3 = \gamma_4 = \gamma_6 = 1e2$, $\gamma_2 = \gamma_5 = 1.0$, $\epsilon = -0.5$ and $\epsilon_{coll} = 0.003$. The model size of $GenSim$ is $\sim$ 1MB. $GenSim$ takes on average $\sim$ 0.9 *secs* [0.6(ARAP)+ 0.3(PEN)] for inferring a 3000 vertices skirt as compared to on average 5 minutes taken by a physics based simulator ArcSim [16], similar time of ArcSim also reported in [2]. The simulation of a garment on a static body pose is invariant to the global body rotation and position. Hence we first front align the target body pose with respect to the template body pose (T-pose) by removing any global rotation and translation w.r.t the T-pose body root joint. Refer to supplementary for details.

4.4. Ablation Study

Losses Justification: It is well understood that Laplacian and the bending losses are important for maintaining the quality of garment surfaces. We study the effect of potential energy loss, pin loss and restrain loss. We train three variants of PEN on a small subset of 10 *random sequences of a short skirt*. **V1:**($\ell_{pe} + \ell_{ben} + \ell_{sm}$), this variant is trained with potential energy loss (ℓ_{pe}), Laplacian smoothing loss (ℓ_{sm}), and bending loss (ℓ_{ben}), **V2:** in addition to all the losses of V1 we add the pin loss (ℓ_{pin}), and in **V3:**, we add restrain loss ((ℓ_{str}) to the V2 losses. We can observe in Fig. 10, in the **V1** version, since the gravitation force was the only force acting on the vertices it leads to the fall down of the garment. The **V2** version holds the garment with the pin vertices, however, the dominating gravitation energy pulls the vertices downwards which results in arbitrary elongation of the edges. The third version **V3** with additional restrain loss, restrict edges to elongate arbitrarily due to the influence of the gravitation force. This demonstrates, all the losses described in the Sec.3.3 are important and necessary to train PEN in an unsupervised manner.

PEN output vs ARAP deformed garment: To demonstrate that PEN adds meaningful displacements to the ARAP deformed garment $\mathbb{G}$ to make it physically plausible, we show a displacement magnitude map in Fig. 11. The garment and the underlying body shape and pose are the same as Fig. 2. The maximum displacement magnitude produced by the PEN is 75mm. We can observe PEN predicts the high magnitude of displacement for the vertices where the garment needs to displace due to the gravitational force, and where it falls on the body surface its magnitude becomes almost close to zero. Also, while producing the displacements for each vertex, PEN maintains the cloth quality. We show the rendered normal map of the template garment and the predicted garment in Fig. 12.

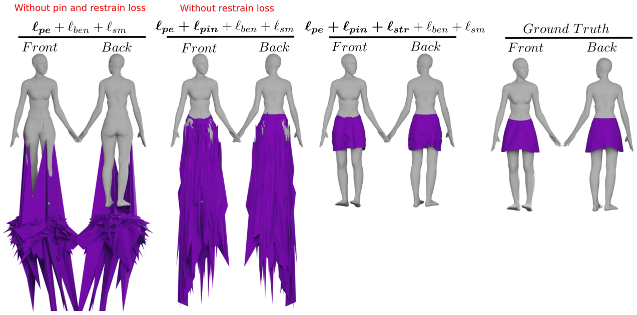
Without pin and restrain loss
Without restrain loss
$\boldsymbol{\ell_{pe}} + \ell_{ben} + \ell_{sm}$
$\boldsymbol{\ell_{pe}} + \boldsymbol{\ell_{pin}} + \ell_{ben} + \ell_{sm}$
$\boldsymbol{\ell_{pe}} + \boldsymbol{\ell_{pin}} + \boldsymbol{\ell_{str}} + \ell_{ben} + \ell_{sm}$
Ground Truth
Front
Back
Front
Back
Front
Back
Front
Back

Figure 10. Ablation study on loss terms.

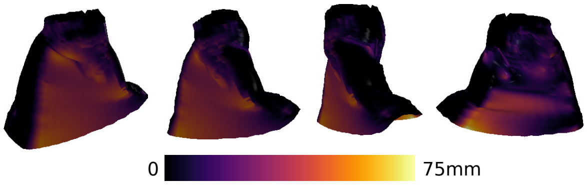

Figure 11. **Displacement Magnitude Map:** Per vertex magnitude of displacement produced by the PEN. Refer to Sec. 4.4 for details

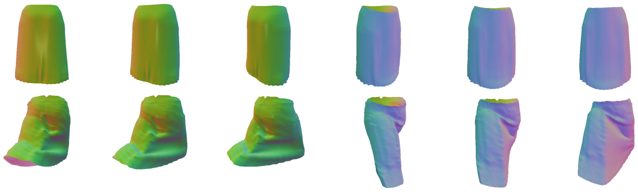

Figure 12. **Normal Map of the Template Garment** (top row) and the predicted garment (bottom row). Refer to Sec. 4.4 for details.

5. Conclusion, Limitations and Future Work

We present the first *unsupervised and generic* garment simulator $GenSim$ that can *simultaneously be trained for multiple types* of garments and generalize to the unseen types of garments of varying sizes, and topologies. Our Motion-aware ARAP initially does a rough alignment of the template garment on the target body pose while avoiding collisions with the body. Then the Physics Enforcing Network adds the correction to the ARAP-deformed garment using unsupervised losses to make it physically plausible. Our results demonstrate generalization on unseen garments, body shapes, and poses. However $GenSim$ has one limitation it doesn't produce temporally consistent garment deformation as the training does not use temporal information. Though there are no significant garment-garment collisions in our results, since we don't explicitly model it, we mention this as another limitation of our method. We intend to overcome the above limitations in the future along with handling layered garments, and garments such as open shirts, jackets etc.

References

[1] Pytorch. https://pytorch.org/. 7

[2] Aric Bartle, Alla Sheffer, Vladimir G Kim, Danny M Kaufman, Nicholas Vining, and Floraine Berthouzoz. Physics-driven pattern adjustment for direct 3d garment editing. *ACM Trans. Graph.*, 35(4):50–1, 2016. 8

[3] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 4, 5

[4] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 1, 2, 6

[5] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Pbns: physically based neural simulation for unsupervised garment pose space deformation. *ACM Transactions on Graphics (TOG)*, 40(6):1–14, 2021. 1, 2, 3, 7

[6] Hugo Bertiche, Meysam Madadi, Emilio Tylson, and Sergio Escalera. Deepsd: Automatic deep skinning and pose space deformation for 3d garment animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5471–5480, 2021. 1, 2, 7

[7] Robert Bridson, Sebastian Marino, and Ronald Fedkiw. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*, pages 3–es. 2005. 6

[8] Y D. Li, Min Tang, Yun Yang, Zi Huang, R F. Tong, SC Yang, Yao Li, and Dinesh Manocha. N-cloth: Predicting 3d cloth deformation with mesh-based networks. In *Computer Graphics Forum*, volume 41, pages 547–558. Wiley Online Library, 2022. 2

[9] Erhan Gundogdu, Victor Constantin, Shaifali Parashar, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet++: Improving fast and accurate static 3d cloth draping by curvature loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):181–195, 2020. 1

[10] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet: A two-stream network for fast and accurate 3d cloth draping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8739–8748, 2019. 2

[11] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11127–11137, 2021. 7

[12] Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–684, 2018. 1, 2

[13] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172, 2000. 2

[14] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2018. 2, 4

[15] Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1205–1211. IEEE, 2019. 4

[16] Rahul Narain, Armin Samii, and James F O'brien. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)*, 31(6):1–10, 2012. 8

[17] Xiaoyu Pan, Jiaming Mai, Xinwei Jiang, Dongxue Tang, Jingxiang Li, Tianjia Shao, Kun Zhou, Xiaogang Jin, and Dinesh Manocha. Predicting loose-fitting garment deformations using bone-driven motion networks. *arXiv preprint arXiv:2205.01355*, 2022. 1, 2, 3

[18] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7365–7375, 2020. 1, 2

[19] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2020. 4

[20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 7

[21] Igor Santesteban, Miguel A Otaduy, and Dan Casas. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, volume 38, pages 355–366. Wiley Online Library, 2019. 1, 2

[22] Igor Santesteban, Miguel A Otaduy, and Dan Casas. Snug: Self-supervised neural dynamic garments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8140–8150, 2022. 1, 2, 3

[23] Igor Santesteban, Nils Thuerey, Miguel A Otaduy, and Dan Casas. Self-supervised collision handling via generative 3d garment models for virtual try-on. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11763–11773, 2021. 1, 7

[24] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007. 2, 3

[25] Rasmus Tamstorf. Derivation of discrete bending forces and their gradients. Technical report, Technical Report, Walt Disney Animation Studios, 2013. 6

[26] Qingyang Tan, Yi Zhou, Tuanfeng Wang, Duygu Ceylan, Xin Sun, and Dinesh Manocha. A repulsive force unit for garment collision handling in neural networks. *arXiv preprint arXiv:2207.13871*, 2022. 1, 2

[27] Lokender Tiwari and Brojeshwar Bhowmick. Deepdraper: Fast and accurate 3d garment draping over a 3d human body. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1416–1426, 2021. 1, 2

[28] Lokender Tiwari and Brojeshwar Bhowmick. Garsim: Particle based neural garment simulator. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4472–4481, 2023. 1, 2

[29] Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum*, volume 39, pages 145–156. Wiley Online Library, 2020. 1

[30] Tuanfeng Y Wang, Duygu Ceylan, Jovan Popović, and Niloy J Mitra. Learning a shared shape space for multimodal garment design. *ACM Transactions on Graphics (TOG)*, 37(6):1–13, 2018. 2

[31] Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. Discrete quadratic curvature energies. *Computer Aided Geometric Design*, 24(8-9):499–518, 2007. 6

[32] Meng Zhang, Duygu Ceylan, and Niloy J Mitra. Motion guided deep dynamic 3d garments. *arXiv preprint arXiv:2209.11449*, 2022. 1, 2, 3