PBNS: Physically Based Neural Simulator for Unsupervised Garment Pose Space Deformation

Hugo Bertiche, Meysam Madadi and Sergio Escalera
University of Barcelona and Computer Vision Center, Spain
hugo_bertiche@hotmail.com

Figure 1: We present a neural simulator for cloth in human-centric scenarios. Our methodology yields skinned models with Pose Space Deformations through an implicit Physically Based Simulation using deep learning framework. This figure shows different neurally simulated outfits on different unseen body poses. Shown results are raw predictions.

Abstract

We present a methodology to automatically obtain Pose Space Deformation (PSD) basis for rigged garments through deep learning. Classical approaches rely on Physically Based Simulations (PBS) to animate clothes. These are general solutions that, given a sufficiently fine-grained discretization of space and time, can achieve highly realistic results. However, they are computationally expensive and any scene modification prompts the need of re-simulation. Linear Blend Skinning (LBS) with PSD offers a lightweight alternative to PBS, though, it needs huge volumes of data to learn proper PSD. We propose using deep learning, formulated as an implicit PBS, to unsupervisedly learn realistic cloth Pose Space Deformations in a constrained scenario: dressed humans. Furthermore, we show it is possible to train these models in an amount of time comparable to a PBS of a few sequences. To the best of our knowledge, we are the first to propose a neural simulator for cloth. While deep-based approaches in the domain are becoming a trend, these are data-hungry models. Moreover, authors often propose complex formulations to better learn wrinkles from PBS data. Dependency from data makes these solutions scalability lower, while their formulation hinders its applicability and compatibility. By proposing an unsupervised methodology to learn PSD for LBS models (3D animation standard), we overcome both of these drawbacks. Results obtained show cloth-consistency in the animated garments and meaningful pose-dependant folds and wrinkles.

1. Introduction

Animation of draped humans has been widely explored by the computer graphics community because of its wide range of potential applications: videogame, film industry, and nowadays, also in virtual and augmented reality VR/AR. We can split the different animation approaches based on their goal: performance or realism. On one hand, Physically Based Simulation (PBS) [6, 18, 26, 27, 33, 36, 39] strategies discretize the space and time to apply basic

physics laws. The realism obtained is closely related to how fine-grained is the discretization. At the same time, the computational cost greatly increase along with level of realism. Moreover, simulation parameters need to be properly fine-tuned in order to obtain the desired results. Therefore, expert knowledge is necessary. On the other hand, Linear Blend Skinning (LBS) [13, 14, 16, 21, 37, 38] and Pose Space Deformation (PSD) [3, 4, 17, 19] techniques require a significantly lower amount of computational resources but compromise the realism of the animation. These strategies are suitable for low-computing environments or applications that demand real-time performance (portable devices and videogames).

Due to its recent success in complex 3D tasks [5, 11, 20, 24, 28, 30, 32], we find deep learning as a promising approach to the garment animation problem. The research community has shown an increasing interest on draped 3D human animation through deep learning during the past few years [1, 2, 7, 8, 9, 15, 25, 31]. Commonly, authors propose learning non-linear PSD models from big volumes of PBS data. Hence, these approaches also demand high computational resources to run the simulations. Another possibility for data gathering is through the use of 4D scans. While this solution allows capturing real data, it is necessary to build expensive and constrained setups. Furthermore, data obtained through scans need to be post-processed to be usable. Supervised deep learning based approaches not only depend on expensive data, but are also bounded by it.

In this paper we propose learning PSD for rigged garments leveraging deep learning framework and formulating the problem as an implicit PBS. By using PBS formulation, we force our models to predict consistent, low-energy configurations of the physical system that cloth and body represent. Doing this allows unsupervised training, and therefore, removes the need of data gathering through expensive simulations or scans. Furthermore, we show that our proposed methodology can yield cloth-consistent PSD in a short amount of time (minutes). By eliminating the need of simulating hundreds or even thousands of sequences, we drastically reduce the time needed from garment design to model deployment. This increases the applicability and scalability of the methodology, broadening the scope of real life scenarios that will benefit from it. The final animated garments show cloth-consistency, pose-dependant wrinkles and temporal coherence for unseen pose sequences. Fig. 1 shows some qualitative samples obtained with the methodology described in this paper. Our main contributions are:

- **Unsupervised PSD Learning.** By enforcing physical consistency during the training of the model, we eliminate the dependency of PBS or scan data. As a consequence, this methodology can be applied to an arbitrary number of garments, body shapes and poses without the computational cost of obtaining data for them.
- **Efficient Training, Deployment and Compatibility.** Most deep based approaches in the current literature propose complex formulations to obtain realistic results. This hinders the training process and posterior deployment. Our proposed methodology yields PSD for LBS models, which is the standard for 3D animation, and therefore, it is automatically compatible with all graphic engines and benefits from the exhaustive optimization for this models. This greatly increases the applicability of the methodology.

The rest of the paper is structured as follows. We review related works in Sec. 2. Then, in Sec. 3 we describe our methodology. Sec. 4 defines the experimental setup. Later, in Sec. 5 results are presented and discussed. We analyze performance in Sec. 6. Finally, we present our conclusions and future work in Sec. 7.

2. State-of-the-art

In the computer graphics community, the garment animation problem has been tackled for decades. Although deep learning has shown significant progress during recent years, one of its main drawbacks in the case of garment animation is the scarcity of available data and the data-hungry nature of deep-based approaches.

2.1. Computer Graphics

PBS (Physically Based Simulation) permits obtaining highly realistic cloth dynamics, usually relying on the *spring-mass* model. The literature on this regard is exhaustive and mainly addresses the efficiency and robustness of the methodology. This is done through simplifications and specialization on constrained scenarios [6, 26, 27, 36]. As another option, authors propose energy-based optimization approaches for an increase in stability and generalization to additional soft-bodies [18]. Some works describe technical improvements to leverage the extra computational power that GPU parallelization yields [33, 39, 34]. Nonetheless, in spite of the increase on efficiency contributed by other works, achieving a high level of realism comes at a great computational expense. When such resources are not available or real-time performance is a must, PBS cannot be applied. To overcome this, **LBS** (Linear Blend Skinning) is used. LBS is the current standard for 3D animation in computer graphics. Objects motion is driven by an skeleton defined as a set of joints. Vertices of the mesh that represents the 3D object are attached to the joints by a set of blend weights. The transformation (rotation, translation and scaling) of each vertex is the weighted sum of the transformations of the joints with the aforementioned blend weights. Commonly, garments are attached to the same

skeleton that controls the 3D body. We can also find an exhaustive research regarding LBS [13, 14, 16, 21, 37, 38]. This approach allows real-time applications, even in low-computing environments, by sacrificing realism, specially on garment domain. Currently, we can find **hybrid** strategies in the industry. Tight parts of the outfits (e.g., t-shirt and trousers) are attached to the body skeleton while other apparel (e.g., capes and long coats) are simulated. This approach is widely used in the videogame industry, as realism is enhanced without an excessive increase on computational requirements.

2.2. Learning-Based Approaches

LBS models achieve real-time performance, nonetheless, linear transformations are usually not enough to capture the motion of soft-tissue objects such as cloth. Furthermore, LBS might suffer of skinning-related artifacts. **PSD** (Pose Space Deformation) aims to address these drawbacks by applying corrective deformations to LBS models before skinning [17]. This helps reducing artifacts and also allows representation of high-frequency details that depend on the pose of the object. Hand-crafted PSD is intractable for complex models (such as the human body) and it is usually learnt from data. Authors have shown that PSD approaches are able to model the human body [3, 4, 19]. Deformations basis are obtained by linear decomposition of hundreds or thousands of 3D body scans. Following this fashion, for garments, Guant et al.[9] propose performing the same computation for synthetic garment data gathered through PBS. Later, Lähner et al.[15] extend the idea by computing the aforementioned linear decomposition against temporal feature arrays processed by a Recurrent Neural Network (RNN), achieving non-linearity w.r.t. the pose. Santesteban et al.[31] explicitly apply a non-linear mapping with a Multi-Layer Perceptron (MLP) for a single fixed garment. While these approaches achieve appealing results, each new garment or outfit requires repeating the simulation and learning process, thus hindering scalability and applicability. Authors commonly address this drawback by leveraging existing body models (like SMPL[19]). Garments are encoded on top of the human body as subsets of displaced vertices [1, 2, 7, 8, 25]. Following the idea of exploiting the human body model, Patel et al.[25] use subsets of body vertices as few different garments to later learn garment-specific models for high frequency cloth details. Bertiche et al.[7] perform a similar encoding for thousands of different garments. This allows learning a continuous space for garment topology. Later they use each garment representation within this space to condition the pose-dependant vertex offsets. Using a body model to represent garments allows handling multiple types with a single model. Nonetheless, huge volumes of data are still needed to train these models. Furthermore, it has been proven that deep neural networks are biased to lower frequencies [29], and as noted by Patel et al.[25], this effect is more significant on cloth domain when training a single model to represent many different garment types. This means that in order to obtain high-resolution garment predictions, it is better to exhaustively simulate and train for individual garments. Our proposed methodology allows skipping the simulation step and efficiently learn, in few minutes, PSD for a given LBS model of a garment or outfit.

3. Neural Cloth Simulation

Classical computer graphics approaches resort to manual template skinning and/or costly simulation, which compromise realism, performance or applicability. On the other hand, learning based approaches, non-deep and deep, propose a manual skinning followed by a data-driven method to compute Pose Space Deformations. Since simulated data is still necessary, a significant computational investment is required for each new garment, body shape or fabric. In addition, models able to handle multiple garment types and body shapes are more biased to lower frequencies (smoother predictions). We propose learning garment-specific (or outfit-specific) PSD unsupervisedly by enforcing physical laws, addressing the main drawbacks of previous works in terms of data requirements.

3.1. Formulation

Our goal is to obtain cloth-consistent PSD for a given garment or outfit rigged to the skeleton of an LBS body model, in order to animate cloth and body at once. The per-vertex formulation of skinning with PSD w.r.t. an articulated skeleton, represented as a set of joints $J \in \mathbb{R}^{K\times 3}$, for a given template garment or outfit in rest pose $\mathbf{T} \in \mathbb{R}^{N\times 3}$ is defined as:

$$t'_i = \sum_{k}^{K} w_{k,i} G_k(\theta, J)(t_i + dt_i(\theta)), \tag{1}$$

where $w_{k,i}$ is the blend weight of vertex i and joint k, G_k is the linear transformation matrix corresponding to joint k, θ is the skeleton pose in axis-angle representation, t_i is the i-th garment vertex in rest pose and dt_i is the pose space deformation corresponding to this vertex. We need to find a valid skinning as a set of blend weights $\mathcal{W} \in \mathbb{R}^{N\times K}$ and a PSD as a mapping $f : \theta \rightarrow \{d\mathbf{T} \mid \mathbf{T}_\theta = \mathbf{T} + d\mathbf{T}\}$ such that the output unposed garment $\mathbf{T}_\theta \in \mathbb{R}^{N\times 3}$ is properly aligned with the body (no collisions) and shows a realistic cloth-like behaviour after skinning. We propose using a neural network to approximate f. Note that our formulation is not dependant on the chosen body model, and the only requirement is to have a database of poses for the body.

Blend Weights. In the current literature we find authors that rely on the assumption that garments closely follow

body motion [31, 7, 25]. Results presented by these works prove it is a valid assumption that allows for a significant simplification of the problem. We rely on this to compute blend weights for our template garment. For each vertex $t_i \in \mathbf{T}$ we assign blend weights equal to those of the closest body vertex, with the body in rest pose (aligned with garment). Skirts break the assumption that cloth and skin are close to each other. For these kind of garments, we allow blend weights to be optimized along with network weights. We observed that doing this increases the model convergence speed. For other types of garments, we see no significant differences, except the computational overhead of optimizing blend weights along with the rest of the trainable parameters.

PSD. Skinning alone is not enough to properly model garments, as cloth behaviour is highly non-linear. For this reason, we formulate the model with PSD. Classical computer graphics approaches rely on linear decomposition from training samples to obtain a PSD matrix like $\mathbf{D} \in \mathbb{R}^{|\theta| \times N \times 3}$, where $|\theta|$ is the dimensionality of the pose array and N is the number of vertices of the 3D mesh to animate. We propose to first obtain a high level embedding of the pose array θ through a neural network as $\mathbf{X} = f_{\mathbf{X}}(\theta)$ and use this embedding to obtain the final deformations with a PSD matrix as $\mathbf{D} \in \mathbb{R}^{|\mathbf{X}| \times N \times 3}$. This allows modelling any non-linear mapping from θ to dT thanks to the universal approximation properties of neural networks. Learning of both $f_{\mathbf{X}}$ and $\mathbf{D}$ is performed following a deep learning framework, where variables are optimized by minimizing a loss function with batches of different input pose samples.

3.2. Architecture

We design $f_{\mathbf{X}}$ as a Multi-Layer Perceptron (MLP). More specifically, 4 fully connected layers with a dimensionality of 256 and ReLU activation function. Then, to obtain the posed outfit $\mathbf{V}_\theta$ for a given θ:

$$\mathbf{V}_\theta = W(\mathbf{T} + f_{\mathbf{X}}(\theta) \cdot \mathbf{D}, \theta, \mathcal{W}) \tag{2}$$

Where $W(\cdot, \theta)$ is the skinning function for pose θ and blend weights $\mathcal{W}$, and the product $f_{\mathbf{X}}(\theta) \cdot \mathbf{D}$ is computed as $\sum_i^{|\mathbf{X}|} f_{\mathbf{X}}(\theta)_i \mathbf{D_i}$. Under this formulation, we can approximate any non-linear mapping from θ to $\mathbf{V}_\theta$ while keeping compatibility with current graphic engines. Fig. 2 shows an overview of the proposed methodology. The input of the model is the pose array θ, which is processed through the aforementioned MLP to yield a high-level pose embedding $\mathbf{X}$. This embedding $\mathbf{X}$ is multiplied with the PSD matrix $\mathbf{D}$ to obtain garment vertex deformations. Both, the MLP and the matrix $\mathbf{D}$ are learnt through training (and blend weights $\mathcal{W}$ are optimized from their initial proximity-based values for outfits with skirt). Finally, the deformed garment (or outfit) is skinned along the body according to θ and blend weights $\mathcal{W}$. For the rest of the paper, for clarity reasons, we consider the PSD matrix D to be part of the network.

3.3. Training

Our main contribution is a methodology to estimate PSD for garment animation without PBS data. Previous approaches rely on linear decomposition or minimization of an error w.r.t. data. Therefore, for each new garment to design, these approaches require data gathering through expensive simulations. Furthermore, learning from data has additional disadvantages. First, the learnt model will be limited by the data distribution, which might be incomplete. Secondly, animation data has not only computational cost for generation, but also for storing. Finally, deep learning models are known to have a bias to lower frequencies [29], which, in this domain, is manifested as overly smoothed and stiff predictions. This compels researchers to develop more complex models with specific formulation to deal with wrinkles[15, 25] and generate more cloth-consistent predictions. This hinders the applicability, scalability and compatibility of the models. With our proposal, we overcome all data-related drawbacks, while ensuring predictions will fulfill the physical properties of cloth.

Just as physical systems are implicitly *optimized* by acting forces $\mathbf{F} = \nabla U$ (where U is the potential energy of the system), we train our neural network under a loss defined as a potential energy, such that our model will learn to predict consistent, low-energy stable configurations. Note that by back-propagating losses through the skinning formulation, the network implicitly learns unposed deformations. We define our global loss (or potential energy) as:

$$\mathcal{L} = \mathcal{L}_{cloth} + \mathcal{L}_{collision} + \mathcal{L}_{gravity} \tag{3}$$

Where $\mathcal{L}_{cloth}$ corresponds to the elastic potential energy of the garment, and will guide the model to predict cloth-consistent meshes. The term $\mathcal{L}_{collision}$ formulates body penetrations as a potential energy, thus its gradients will push cloth vertices to valid locations. Finally, $\mathcal{L}_{gravity}$ is the gravitational potential energy, which will minimize vertices height within the constraints set by the other loss terms.

Cloth consistency. The first term of our loss is related to the cloth consistency of the predictions. That is, we want the output meshes to fulfill certain properties we find in cloth. Classical computer graphics approaches usually rely on the mass-spring model to simulate cloth. We extend this idea to deep learning by designing a cloth loss term as:

$$\mathcal{L}_{cloth} = \lambda_e \mathcal{L}_{edge} + \lambda_b \mathcal{L}_{bend} = \lambda_e \left\| E - E_{\mathbf{T}} \right\|^2 + \lambda_b \Delta(\mathbf{N})^2, \tag{4}$$

where $E \in \mathbb{R}^{N_E}$ are the predicted edge lengths, $E_{\mathbf{T}} \in \mathbb{R}^{N_E}$ are the edge lengths on the rest garment $\mathbf{T}$ (with N_E as the number of edges in the mesh), $\mathbf{N} \in \mathbb{R}^{N_F \times 3}$ are the

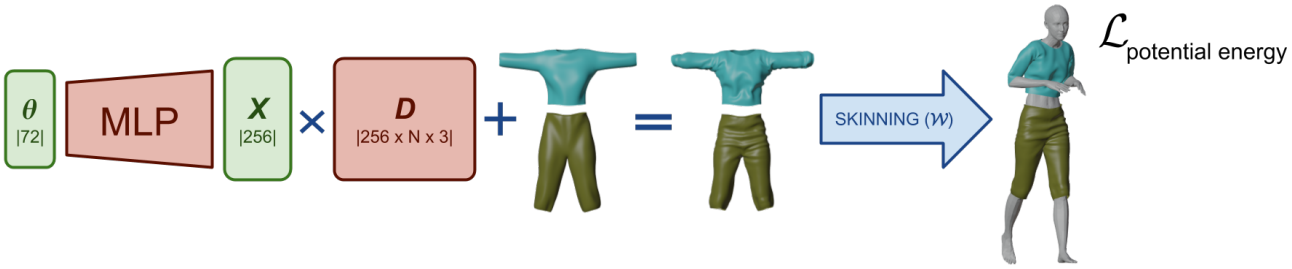

Figure 2: Methodology overview. Pose parameters are fed to a 4-layer Multi-Layer Perceptron with 256 dimensions in each layer. The output high-level pose embedding is multiplied with a PSD matrix $\mathbf{D}$ to obtain deformations for the garment/outfit. As standard, deformations are applied in rest pose garments and skinned along with the body (note that garment blend weights are assigned by proximity w.r.t. the body in rest pose). Finally, we train by optimizing the potential energy of the output physical system represented by the body and the cloth. The parts of the model in red correspond to the trainable parameters.

face normals (for N_F triangular faces), $\Delta(\cdot)$ is the Laplace-Beltrami operator and λ_e and λ_b are balancing factors. The term $\mathcal{L}_{edge}$ ensures that cloth is not excessively stretched or compressed. It is formulated as the potential elastic energy of the system, such that its gradients act as forces. On the other hand, $\mathcal{L}_{bend}$ will enforce locally smooth surfaces by penalizing differences between neighbouring face normals (hinge-like forces). Note that the latter is computed taking into account face connectivity, not vertex.

Collisions. Next, the model needs to handle collisions with the body model. To do so, we design the following loss:

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

where A represents the set of correspondences (i, j) between predicted outfit and body, respectively, through nearest neighbour, $\mathbf{d}_{j,i}$ is the vector going from the j-th vertex of the body to the i-th vertex of the outfit, $\mathbf{n}_j$ is the j-th vertex normal of the body, ϵ is a small positive threshold to increase robustness and λ_c is a balancing weight ($\epsilon = 4mm$ and $\lambda_c = 25$ in our experiments). This loss is crucial to obtain valid predictions and its gradients will push outfit vertices outside the body. It is designed under the assumption that cloth closely follows the skin, which we can safely assume given that initial skinning blend weights are assigned by proximity. Note that, as with PBS, invalid bodies (self-collided) will corrupt the results. While this is a naive implementation of a collision loss, it works well in practice and similar L1 formulations have already been used in deep-based approaches [35, 12, 10]. We opted for a quadratic term to avoid generalization and stability issues, plus, it helps achieving a balance w.r.t. the other loss terms.

Gravity. We include an additional term to enforce more realistic garment predictions. This term will model the effect of the gravity. From classical mechanics, we know that potential gravitational energy is $U = m \cdot g \cdot h$, where m is the mass of the object, g is the gravity and h is the height of the object. Since m and g are constant, we can understand this loss as $\mathcal{L}_{gravity} = k\mathbf{V}_{\theta z}$. In other words, we are minimizing the Z coordinate (vertical axis) of each vertex of the predicted garments.

Pinning. For some garments we want certain vertices not to move around. For example, lower body garments will usually fall down as training progresses due to the gravity loss. We want to restrict waist vertices deformation such that it remains attached to its original position. The concept of pinning appears in most cloth simulators. To this end, we implement an L2 regularization loss on the deformations dt of each vertex defined as pinned down. Note that a hard constraint would most likely produce collisions against the body, so vertices need to be able to move slightly. Then, we include it in the loss as an extra term with a balcing weight $\lambda_{pin} = 10^3$.

By formulating both $\mathcal{L}_{cloth}$ and $\mathcal{L}_{gravity}$ as physical magnitudes, their corresponding loss balancing weights are directly related to the properties of the fabric we want to simulate: Young's modulus for the elasticity and its mass for gravity. This provides of explainability to the approach. The rest of the losses, as with classical computer graphics PBS, are simplifications of the underlying physics.

4. Experiments

In this section we will first describe the process to apply the methodology explained in Sec. 3 and the metrics used for monitoring the training progress. Then, we show and evaluate qualitative results as well as other interesting properties of our neural simulator.

4.1. Body model

SMPL [19] is the current standard in the literature for human analysis and garment animation. This model is an LBS with PSD obtained through thousands of accurate 3D scans of different subjects. Its underlying skeleton is defined as

a set of $K = 24$ joints. Public pose databases are available for this model (AMASS [22]). We then choose SMPL for the experimental part because both model and pose data are available to the public. Nonetheless, the methodology described in this paper is compatible with any 3D model rigged to an skeleton. SMPL also allows generating different body shapes through blend shapes. During neural simulation, body shape is fixed (just as with PBS, where, in general, we do not want the body shape to change during simulation).

4.2. Template outfit

Once a body model is selected, a garment or outfit is designed for the body in rest pose, with an approximate resolution of 1cm in our experiments. We smooth templates as much as we can before neural simulation. This will ensure that high frequency details and deformations are indeed generated by the model from the pose. Then, initial blend weights for the cloth are obtained by proximity to the body.

4.3. Pose database

As mentioned before, neural simulation requires a database of valid poses for the selected body model. We define as valid poses those that do not produce self-collisions when applied to the 3D body model. As with regular PBS, neurally simulating cloth over bodies with self-collisions will generate inconsistent repelling forces and corrupt the results. For SMPL, we have $|\theta| = 3K = 72$. We choose CMU MoCap pose sequences. This dataset contains 2667 pose sequences of different length, performed by different subjects. It totals around 4.3M individual poses. We split the database into train and test per subject, thus ensuring no subject or sequence is repeated in both sets, as $85/15\%$. Then, to ensure pose balance, we randomly sample $N = 3000$ poses from the training set, such that not any pair of poses have a distance $d < 0.5$, with $d = max(|\theta_i - \theta_j|)$. Thus, for any two poses, there is at least one parameter with a difference equal or bigger than 0.5 radians (we omit global orientation for this sampling). Later, we split into training and validation set as $85/15\%$. Quantitative and qualitative results presented in this paper correspond to unseen test poses.

4.4. Metrics

There is no PBS data to compare with, so we define metrics to monitor the training progress. Note that these metrics are computed on the validation set during training and finally on test set. They are as follow:

- **Edge.** Compression or elongation distance for outfit mesh edges.
- **Bending.** Cosine distance between neighbouring face normals.

Table 1: Proposed metrics to track the training progress. Computed for the test set. For collision, lower means better. The rest are used to guide the learning and identify neural simulation failures. Gravitational potential energy will differ greatly depending on pose distribution, garment style, mass and definition of 0 energy, therefore, we omit this metric from the table.

Edge (mm)	Bend (rad.)	Collision	Gravity
0.25 - 1	0.02 - 0.05	0.5% - 0.75%	-

- **Collision.** Ratio of collided vertices.
- **Gravity.** Potential gravitational energy.

Except for collisions, a zero valued metric does not mean perfect predictions. For edges, we know cloth stretches and compresses as needed to fit the environment, and it is highly unlikely to find clothes without stored elastic energy. The same thing happens for the bending metric. Gravity metric should slowly decrease until it reaches a minimum. Therefore, in practice, these metrics are helpful to guide the training progress, fine-tune its hyperparameters and balance losses. Tab. 1 shows typical metric values for realistic, cloth-consistent predictions. The proposed network is able to keep the edge error (w.r.t. rest outfit) below a millimeter. For bend metric, a value in the range $[0.02, 0.05]$ provides with smooth surfaces without excessive flattening (note that the optimal value depends on the resolution of the outfit meshes, 1cm in our experiments). Then, we observe the model is able to obtain almost collision-free predictions for unseen poses. This point is crucial for the applicability of our model in real scenarios. Finally, gravity metric depends on the definition of 0 potential energy (usually at height $z = 0$), the mass of the outfit and the pose distribution. This means its absolute value, or even the difference as training progresses, is not very informative. In spite of that, edge, collision and gravity metrics are the main indicators that simulation has failed due to unbalanced losses.

5. Results

We test our methodology with different outfits and different bodies. Since ours is an unsupervised approach, a quantitative evaluation of the results is not possible.

5.1. Qualitative

For a qualitative evaluation of the results, we refer first to Fig. 1. In this image we show a few samples with different pose, outfit and, one of them, different body. As it can be seen, our learnt PSD can generate appropriate wrinkles around bent joints in a realistic manner to fulfill the energy balance requirements imposed by our loss during training. Then, for a more in-depth analysis, we refer to

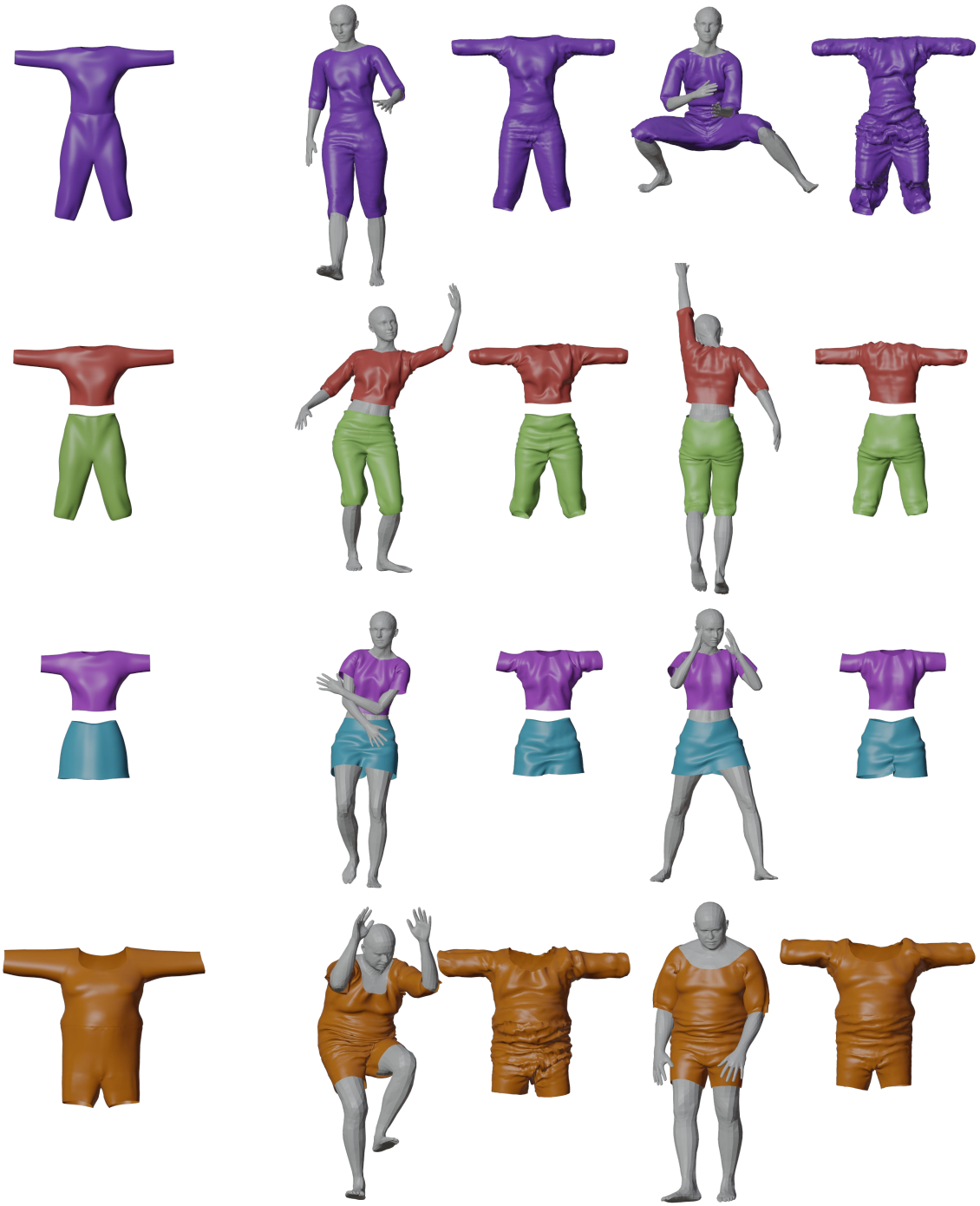

Figure 3: Qualitative results. Leftmost, we depict the template outfit of each row. Note how templates are smoothed as much as possible. Then, for each row we visualize the output for two unseen test poses. For each sample, we show the output dressed human (left) along with its corresponding deformed outfit in rest pose (right). As it can be seen, the learnt Pose Space Deformations generate folds and wrinkles to fulfill the energy balance requirement of the physical system. The last row depicts results of simulation with a different body shape (and its appropriately aligned template outfit) to show generalization to different bodies. All of the shown samples were obtained after just a few minutes of training.

Figure 4: Textured qualitative samples. This rendering gives a more approximate idea to the final application-level looks that our methodology yields. These results correspond to neural simulations of 3 additional outfits on top of 3 different bodies, further showing the generalization capabilities of our methodology.

Fig. 3. Here we show, on one hand, the template outfit of each sample. Templates are smoothed as much as possible. Later, for each template, the output for two extra unseen poses are shown (different from Fig. 1). For each of these samples, the final rigged draped human is visualized (left) along with the unposed deformed template garment $\mathbf{T}_\theta$ (right). Templates show deformations to satisfy the energy constraints which would not be possible with skinning alone. From the first row, we can notice how big deformations are due to collisions for extreme poses. Also, while deformed template can look noisy, it looks realistic after skinning. On the second row we can see deformations on the back of the outfit. Nonetheless, as human bodies usually bend forward, wrinkles in the front are more evident. The third row shows samples with a skirt. Note how in the second sample, the skirt deformations need to correct rotations due to leg movements (deformed template has a discontinuity). For outfits with skirt, we allow optimization of blend weights, along with the rest of the network, to alleviate the correction required due to leg motion. In spite of this, the effect is not fully mitigated. Finally, the last row shows results obtained with a different body shape. As aforementioned, the methodology presented in this paper is compatible with any 3D model rigged to an skeleton. On Fig. 4 we rendered more qualitative results for different bodies and outfits. All of these visualizations correspond to models trained during just a few minutes without any post-processing.

5.2. Controllable Parameters

Standard computer graphics simulators include the possibility of tuning simulation parameters. By formulating our learning process as an implicit PBS, we establish a direct relation between loss terms and physical properties of cloth. More specifically, the loss balancing weights of $\mathcal{L}_{edge}$ and $\mathcal{L}_{bend}$ allows simulating different *fabrics*, with different elasticity and flexibility. Fig. 5 shows the result of neurally simulating the same garment for the same body and with the same training poses, but different loss balancing weights. From left to right, we have values $\lambda_e = 5, 10, 15, 20, 25$ and $\lambda_b = 1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}$. As it can be observed, lower weights for these losses generate finer wrinkles, while higher values will produce stiffer cloth behaviour. This control over the output of our neural simulator represents an advantage over supervised learning

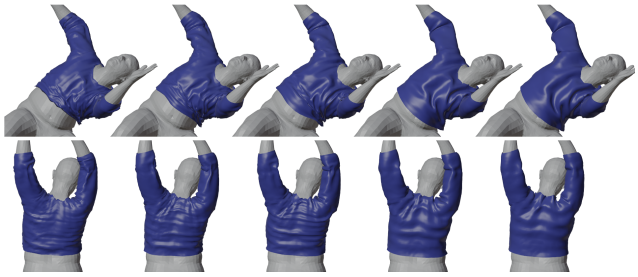

Figure 5: Formulating the training process as a PBS allows to control cloth behaviour by changing parameters related to its physical properties. From left to right, we increase the balancing weights of loss terms $\mathcal{L}_{edge}$ and $\mathcal{L}_{bend}$. As it can be seen, the garment deformations obtained change consistently with the balancing weights (the higher weight, the stiffer the output).

Figure 6: Initial vs optimized blend weights. We show the effect of optimizing blend weights for the skirt. The three outfits at left show the distribution of the initial blend weights for root joint, left hip and right hip respectively. The three outfits on the right show the distribution of the same blend weights after training.

based approaches, which are limited to learn the behaviour present in the data.

5.3. Blend weights

A proximity based skinning works well when the outfit fulfills the requirement that it closely follows the body. For the case of skirts, this assumption is broken. Due to its topology, proximity blend weights show a discontinuity between left and right side. Fig. 6 shows a visualization of this. We render the blend weights distribution for the first three joints (root, left hip and right hip respectively). At left, we show the initial blend weights. As it can be seen, there is a sudden change in color in the middle of the skirt. At the right side of the figure, we render the same visualization after optimizing blend weights during training. We observe how the sudden jump from left to right leg has been smoothed to the point that we find non-zero blend weights for the three depicted joints in the middle of the skirt. Note that this skinning is later complemented with the learnt PSD to satisfy the energy stability.

5.4. Comparison

To the best of our knowledge, we are the first to propose an unsupervised approach to learn PSD for garment animation. Then, we compare ourselves in a qualitative way against other deep learning based approaches. Tab. 2 summarizes the main differences and advantages our approach presents. First, as aforementioned, no other related work is able to learn without PBS data.

From these works, ours is the only one that presents its formulation in a compatible way w.r.t. the standards in 3D animation. Most graphic engines work with blend shapes, which allow morphing or interpolating between different mesh shapes. Blend shapes are formulated as a feature array $\mathbf{X}^F$ and a matrix with different deformation basis as $\mathbf{D}^{F\times N\times 3}$ (where N is the number of vertices). Pose Space Deformation is an specific case of blend shapes, in which the feature array depends directly on the pose. Therefore, since we propose learning a matrix like $\mathbf{D}$ that shall yield vertex deformations when multiplied with a high-level pose embedding array $\mathbf{X}$, our approach is directly compatible with current 3D animation pipelines. This is not the case for the other approaches, since they use more complex architectures or specific methodologies to deal with wrinkles. Again, this is a drawback related to learning from data. Since neural networks are biased towards lower frequencies, authors resort to more complex formulations to mitigate this effect.

Obtaining PBS data requires a significant investment on computational resources. Also, deep learning is a data-hungry approach. Furthermore, some of these previous works require an expensive data formatting and preprocessing (usually registration against a body or template, plus unposing each sample), which in turn, might introduce errors in the data. Due to this, the time from outfit design (in rest pose) until final model deployment can take hours at best, and usually days. Since we do not require data, nor pre-processing of any kind, only the training time separates outfit design from deployment. This greatly increases the applicability and scalability of the model, since its training time allows for fast integration of new 3D models (e.g.: new outfits for a videogame or virtual try-on database, or faster pre-visualization times for CGI artists).

Since we formulate our learning process as an implicit PBS, there is a direct relation between loss terms and the physical properties of the cloth. As in standard cloth simulators, we can set some fabric properties, such as the modulus of Young, bending resistance (mass-spring model) or its mass. Additionally, we can control the relative position of certain vertices through *pinning*. We use this property for the waist vertices of 2-pieces outfits. For the rest of the approaches, and again due to their dependency on data, cloth parameters cannot be tuned, and results will, at most, mimic the dataset distribution.

Finally, we observed how some methodologies try to leverage the body model mesh (or UV map) to encode different garment types or topologies as subsets of vertices. While this allows for generalization to different garments with the same model, it hinders its representative capabilities. These kind of approaches are unable to deal with overlapping layers of cloth or outfits with complex geometric details. Fig. 7 shows a simple example of a garment that these kind of approaches cannot deal with.

6. Performance

We train our model on our subset of 2550 training poses with a batch size of 16 and Adam optimizer. We run our experiments on a GTX1080Ti. It takes around $1-2$ minutes per epoch, depending on the amount of collisions against the body. Since there is no quantitative error, the stop criterion consists on qualitatively assessing validation predictions. On our experiments, we observed that a single 1 minute epoch is enough to start showing visually appeal-

Table 2: Comparison against other deep learning based approaches. We are the first to propose a completely unsupervised methodology. Additionally, we formulate our model as the standard in 3D animation, allowing automatic compatibility with most graphic engines. Since we skip data generation (or gathering) plus expensive formatting or pre-processing, our model can be trained and deployed in a matter of minutes. By formulating our problem as a PBS, we make our loss explainable and allow control over the output. By training outfit-specific models (as opposed to extending a body model), we are not bound in terms of representativity. Finally, we will make our code available for future research.

	Requires PBS data	Compatibility	Simulation + Training time	Explainability and controllability	Arbitrary topologies	Public Code
Santesteban et al.[31]	Yes	No	Hours/Days	No	Yes	No
DeepWrinkles[15]	Yes	No	Hours/Days	No	No	No
DRAPE[9]	Yes	No	Hours/Days	No	Yes	No
GarNet[10]	Yes	No	Hours/Days	No	Yes	No
CLOTH3D[7]	Yes	No	Hours/Days	No	No	No
TailorNet[25]	Yes	No	Hours/Days	No	No	Yes
PBNS (Ours)	**No**	**Yes**	**Minutes**	**Yes**	**Yes**	**Yes**

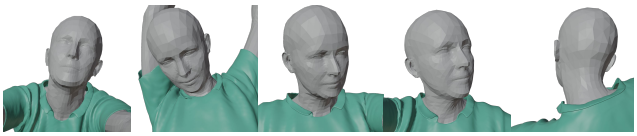

Figure 7: This figure shows an example of a neurally simulated garment with a collar. Some previous works try to leverage the body model to represent multiple garment styles in a single network. Such works are unable to represent this kind of details (overlapping cloth on body projection).

ing, cloth-consistent predictions with pose-dependant wrinkles. Nonetheless, it might take from 10 to 50 epochs to converge, depending on outfit complexity and body shape.

The resulting trained model consists on an LBS model with PSD. These kind of models are the current standard for 3D animation and, therefore, graphic engines are exhaustively optimized for them. This means it is likely to be the optimal formulation in terms of performance. We consider the analysis of LBS models with PSD to be out of the scope of this paper. Nevertheless, we have to account for the computational cost of the MLP. If we consider applications such as videogames or movie character design, high-level pose embeddings can be computed beforehand, which means 0 extra computational cost. On the other hand, on scenarios where pre-processing pose arrays θ is not possible, its cost must be included into the animation performance. Our MLP is designed with 4 layers of 256 dimensions each with ReLU activations. The number of operations for a fully connected layer with input dimensionality F_i and output dimensionality F_o is:

$$O_{FC} = C_{matrix} + C_{bias} + C_{ReLU} = (2F_i + 1)F_o, \quad (6)$$

where the first term corresponds to the matrix multiplication, the second term is the operations for adding the bias term and the final term corresponds to ReLU operations. For the proposed MLP, this is $O_{MLP} = 431104$ operations. On the other hand, we have the operations related to PSD and LBS. First, PSD is obtained by multiplying the feature embedding $\mathbf{X} \in \mathbb{R}^{256}$ with the matrix $\mathbf{D} \in \mathbb{R}^{256\times N\times 3}$. This is:

$$O_{PSD} = C_{products} + C_{reduction} + C_{deformations} = 1536N, \quad (7)$$

Where N is the amount of outfit vertices, $C_{products}$ is the amount of multiplications, $C_{reduction}$ is the amount of additions required to reduce the first axis and $C_{deformations}$ is the number of operations for applying the deformations to the template outfit $\mathbf{T}$. Then, we must consider the cost of computing the blending of the linear transformation matrices $G_i \in \mathbb{R}^{4\times 4}$ for each vertex. This is:

$$O_{blend} = C_{products} + C_{reduction} = 16N(2K - 1), \quad (8)$$

Finally, the cost of each vertex transformation (note that the mesh has to be extended with a column of 1s, therefore, $\mathbf{T}_\theta \in \mathbb{R}^{N\times 4}$). That is:

$$O_{skinning} = C_{products} + C_{reduction} = 28N, \quad (9)$$

Therefore, the final number of operations for our LBS model with PSD is $O_{LBS+PSD} = (1548 + 32K)N$. This means that for SMPL ($K = 24$) and outfits with vertices in the range of $N \in [10000, 20000]$, the ratio of operations $r = C_{MLP}/C_{LBS+PSD}$ is around $1-2\%$. Since fully connected layers are computed with matrix multiplications and additions, we can assume the same exhaustive optimization as for LBS plus PSD models, and therefore, we can compare their performances by comparing the number of operations. With the given ratio of operations, and taking into account that LBS plus PSD models can achieve hundreds or thou-

sands of executions per second, we can safely assume that our proposed MLP has no impact on the performance.

7. Conclusions, Limitations and Future work

We presented the first unsupervised deep learning based approach for garment animation. More specifically, we described a methodology to automatically obtain Pose Space Deformations for Linear Blend Skinning models, allowing an easy integration into any current 3D animation pipeline. We enabled unsupervised learning by formulating our problem as an implicit Physically Based Simulation. Furthermore, our proposed approach can be trained in a matter of minutes. Therefore, the time from outfit design until model deployment is drastically reduced compared to previous approaches. This gives the methodology a broader applicability and higher scalability. CGI artists can design new animated draped characters more efficiently, and both, videogames and virtual try-ons, can easily introduce new 3D animated models for their outfit databases.

In our approach, neither the input nor the potential energy formulation take into account the temporal dimension. This means that the learnt mapping from pose to mesh is unique. One can easily see how this is not true by imagining simulating the same pose sequence at different speeds. A given pose θ on the sequence shall produce different meshes $\mathbf{V}_\theta$. We believe that including temporal behaviour in our neural simulator is a promising research line as future work. Additionally, garments as skirts or dresses show much more complex dynamics than tight outfits, specially the longer they are. To properly model such clothes it is a requirement to handle the temporal dimension.

Another relevant issue is multi-layered cloth. Standard PBS approaches can rely on proximity repelling forces for self-collision solving, as long as the time step discretization is small enough. On the other hand, a proximity approach in our methodology might yield inconsistent gradients when self-collision appears, since there is not temporal continuity between samples or the way that our model learns. An interesting future work can be dealing with self-collisions for multiple-layers of cloth by following the idea presented in [23]. Volumetric *air* meshes are defined in-between cloth layers and their signed volume is included into the energy formulation to be optimized. Negative air mesh volumes indicate penetration between layers of cloth. Since it is formulated as a static (non-temporal) energy term, it is compatible with our approach.

Finally, the idea of unsupervisedly learning to predict stable and physically consistent systems opens the possibility to generalize this methodology to handle other soft-tissue bodies. For example, hair, or human body self-collisions.

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

References

[1] Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Video based reconstruction of 3d people models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8387–8397 (2018) 2, 3

[2] 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) 2, 3

[3] Allen, B., Curless, B., Popović, Z.: Articulated body deformation from range scan data. ACM Transactions on Graphics (TOG) **21**(3), 612–619 (2002) 2, 3

[4] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH 2005 Papers, pp. 408–416 (2005) 2, 3

[5] Arsalan Soltani, A., Huang, H., Wu, J., Kulkarni, T.D., Tenenbaum, J.B.: Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1511–1519 (2017) 2

[6] Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. pp. 43–54 (1998) 1, 2

[7] Bertiche, H., Madadi, M., Escalera, S.: Cloth3d: Clothed 3d humans. arXiv preprint arXiv:1912.02792 (2019) 2, 3, 4, 10

[8] 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) 2, 3

[9] Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., Black, M.J.: Drape: Dressing any person. ACM Transactions on Graphics (TOG) **31**(4), 1–10 (2012) 2, 3, 10

[10] 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: Proceedings of the IEEE International Conference on Computer Vision. pp. 8739–8748 (2019) 5, 10

[11] Han, X., Gao, C., Yu, Y.: Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. ACM Transactions on graphics (TOG) **36**(4), 1–12 (2017) 2

[12] Jiang, B., Zhang, J., Hong, Y., Luo, J., Liu, L., Bao, H.: Bcnet: Learning body and cloth shape from a single image. arXiv preprint arXiv:2004.00214 (2020) 5

[13] Kavan, L., Collins, S., Žára, J., O'Sullivan, C.: Geometric skinning with approximate dual quaternion blending. ACM Transactions on Graphics (TOG) **27**(4), 1–23 (2008) 2, 3

[14] Kavan, L., Žára, J.: Spherical blend skinning: a real-time deformation of articulated models. In: Proceedings of the 2005 symposium on Interactive 3D graphics and games. pp. 9–16 (2005) 2, 3

[15] 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) 2, 3, 4, 10

[16] Le, B.H., Deng, Z.: Smooth skinning decomposition with rigid bones. ACM Transactions on Graphics (TOG) **31**(6), 1–10 (2012) 2, 3

[17] Lewis, J.P., Cordner, M., Fong, N.: 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. pp. 165–172 (2000) 2, 3

[18] Liu, T., Bouaziz, S., Kavan, L.: Quasi-newton methods for real-time simulation of hyperelastic materials. ACM Transactions on Graphics (TOG) **36**(3), 1–16 (2017) 1, 2

[19] 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), 1–16 (2015) 2, 3, 5

[20] Madadi, M., Bertiche, H., Escalera, S.: Smplr: Deep learning based smpl reverse for 3d human pose and shape recovery. Pattern Recognition p. 107472 (2020) 2

[21] Magnenat-thalmann, N., Laperrire, R., Thalmann, D., Montréal, U.D.: Joint-dependent local deformations for hand animation and object grasping. In: In Proceedings on Graphics interface '88. pp. 26–33 (1988) 2, 3

[22] Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5442–5451 (2019) 6

[23] Müller, M., Chentanez, N., Kim, T.Y., Macklin, M.: Air meshes for robust collision handling. ACM Transactions on Graphics (TOG) **34**(4), 1–9 (2015) 11

[24] Omran, M., Lassner, C., Pons-Moll, G., Gehler, P., Schiele, B.: Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In: 2018 international conference on 3D vision (3DV). pp. 484–494. IEEE (2018) 2

[25] 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) 2, 3, 4, 10

[26] Provot, X.: Collision and self-collision handling in cloth model dedicated to design garments. In: Computer Animation and Simulation'97, pp. 177–189. Springer (1997) 1, 2

[27] Provot, X., et al.: Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In: Graphics interface. pp. 147–147. Canadian Information Processing Society (1995) 1, 2

[28] Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017) 2

[29] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019) 3, 4

[30] Richardson, E., Sela, M., Kimmel, R.: 3d face reconstruction by learning from synthetic data. In: 2016 fourth international conference on 3D vision (3DV). pp. 460–469. IEEE (2016) 2

[31] 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) 2, 3, 4, 10

[32] Socher, R., Huval, B., Bath, B., Manning, C.D., Ng, A.Y.: Convolutional-recursive deep learning for 3d object classification. In: Advances in neural information processing systems. pp. 656–664 (2012) 2

[33] Tang, M., Tong, R., Narain, R., Meng, C., Manocha, D.: A gpu-based streaming algorithm for high-resolution cloth simulation. In: Computer Graphics Forum. vol. 32, pp. 21–30. Wiley Online Library (2013) 1, 2

[34] Tang, M., Wang, T., Liu, Z., Tong, R., Manocha, D.: I-cloth: Incremental collision handling for gpu-based interactive cloth simulation. ACM Transactions on Graphics (TOG) **37**(6), 1–10 (2018) 2

[35] Tiwari, G., Bhatnagar, B.L., Tung, T., Pons-Moll, G.: Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. arXiv preprint arXiv:2007.11610 (2020) 5

[36] Vassilev, T., Spanlang, B., Chrysanthou, Y.: Fast cloth animation on walking avatars. In: Computer Graphics Forum. vol. 20, pp. 260–267. Wiley Online Library (2001) 1, 2

[37] Wang, R.Y., Pulli, K., Popović, J.: Real-time enveloping with rotational regression. In: ACM SIGGRAPH 2007 papers, pp. 73–es (2007) 2, 3

[38] Wang, X.C., Phillips, C.: Multi-weight enveloping: least-squares approximation techniques for skin animation. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation. pp. 129–138 (2002) 2, 3

[39] Zeller, C.: Cloth simulation on the gpu. In: ACM SIGGRAPH 2005 Sketches, pp. 39–es (2005) 1, 2