GSNet: Generating 3D Garment Animation via Graph Skinning Network

Tao Peng
Wuhan Textile University
Wuhan, China
pt@wtu.edu.cn

Jiewen Kuang
Wuhan Textile University
Wuhan, China
jiewen_kuang@163.com

Jinxing Liang*
Wuhan Textile University
Wuhan, China
jxliang@wtu.edu.cn

Xinrong Hu
Wuhan Textile University
Wuhan, China
hxr@wtu.edu.cn

Ping Zhu
Wuhan Textile University
Wuhan, China
zhuping@wtu.edu.cn

Lijun Li
Ningbo Cixing Co.,Ltd
Ningbo, China
lilijun@seu.edu.cn

Feng Yu
Wuhan Textile University
Wuhan, China
yufeng@wtu.edu.cn

Minghua Jiang
Wuhan Textile University
Wuhan, China
minghuajiang@wtu.edu.cn

Figure 1. We propose a approach for generating garment animations that can handle garments with multiple topologies and maintain realistic effects and performance.

Abstract

The goal of digital dress body animation is to produce the most realistic dress body animation possible. Although a method based on the same topology as the body can produce realistic results, it can only be applied to garments with the same topology as the body. Although the generalization-based approach can be extended to different types of garment templates, it still produces effects far from reality. We propose GSNet, a learning-based model that generates realistic garment animations and applies to garment types that do not match the body topology. We encode garment templates and body motions into latent space and use graph convolution to transfer body motion information to garment templates to drive garment motions. Our model considers temporal dependency and provides reliable physical constraints to make the generated animations more realistic. Qualitative and quantitative experiments show that our approach achieves the performance of state-of-the-art dressed body animation.

1. Introduction

Generating realistic-dressed body simulations has been a popular area of research for decades. It can be used in various applications, including computer animation, special effects, virtual try-on applications, the fashion industry, video games, and VR. The most common approached used in the past was a physics-based approach [24, 4, 19, 27, 28, 33, 34, 40], and with the development of deep learning, learning-based approaches [7, 6, 20, 26, 31, 39, 3, 11, 21, 25, 29, 30, 32] are gradually being applied to this field.

The physics-based approach [24, 4, 19, 27, 28, 33, 34, 40] treats the garment simulation as a deformable modeling problem and then uses kinematic computational methods to solve the problem. Finally, accurate simulations are achieved through collision processing. This approach produces high-fidelity simulations. Although it can be accelerated by exploiting the parallelism [12, 23] of GPUs, this approach is mainly limited to offline simulations. Subsequently, the linear skinning approach [14, 15, 18, 22, 36, 37] emerged, connecting each garment mesh vertex to the skeleton by a set of blend weights used for linear combinatorial joint transformations. The garment is attached to the skeleton that drives the body's motion, and the body motion and the garment motion are driven simultaneously through the skeleton motion. This approach has also been widely investigated.

Recently, some learning-based approaches [7, 6, 20, 26, 31, 39, 3, 11, 21, 25, 29, 30, 32] have been proposed, most of which encode the garment as the same topology as the body and then transform the problem to solve the pose space deformations, and finally generate the corresponding animations by a linear transformation. There are also approaches [7, 26, 39, 3] based on supervised learning, which can be extended to templates of garments with multiple topologies. All these approaches transform the problem into solving pose space deformation and then using linear skinning approaches. Although these approaches increase the speed of the simulation significantly, however, the motion of the garment is highly nonlinear, and these approaches lead to much less realistic results.

In this paper, we propose a graph-based neural network model GSNet. The state-of-the-art approaches is to learn the pose space deformation from the data and then solve the problem with a linear skinning approach. In contrast, our model is based entirely on neural networks to generate costume animations. We encode garment templates and human motion into latent space, then translate the human motion information into garment motion by graphical convolution to generate garment motion consistent with human motion without linear skinning. Our model is nonlinear so that we can obtain realistic garment animation effects. In addition, to compensate for the physical inconsistency in supervised learning, we define relevant physical constraints and use them for training the model to predict configurations that satisfy the physical constraints. We also introduce temporal dependency in the model to make the animation more realistic. In addition, our approach can be applied to clothing templates with multiple types of topologies that are inconsistent with the human body topology (see Figure 1 for some examples), and our main contributions can be summarized as follows:

- **Robustness:** Since garment motion is highly nonlinear, using linear skinning to solve it leads to a significant loss of realism when applied to garments. Our model is entirely learning-based, so it is highly nonlinear, which can significantly improve our performance. Moreover, our model maintains good stability after experimental comparison.

- **Physical Consistency:** Supervised learning does not guarantee physical constraints, so we add a physically constrained network after supervised learning to correct for garment vertices that do not conform to the physical constraints. Then we also incorporate the relevant physical constraints. The final generated result conforms to a realistic effect.

- **Multiple Topologies:** Most approaches encode garments to the same topology as the body. When these methods are applied to garment templates that do not have the same topology as the body, the results are very unrealistic. To obtain realistic results, even for garments with different topologies from the body, we recommend encoding the garments into latent space before generating the animation. This can support multiple topologies types of garments, greatly enhancing scalability.

- **Temporal Dependency:** The state-of-the-art approaches are based on generating animations from a single frame, and this approach does not consider the connection between motions. Our network introduces temporal dependence, similar to recurrent neural networks, where we use the output of the current frame of the garment as the input of the next frame. This design allows the result to produce a more plausible wrinkles effect than other single-frame-based animations.

2. Related Work

In this section, we briefly overview prior work on garment animation using computer graphics and learning-based approaches.

2.1. Computer Graphics

Realistic garment animations can be obtained through physics-based simulations, usually through the well-known

mass-spring model. Much research in this area has focused on improving the efficiency and stability of simulations by simplifying or specializing in specific setups [4, 24, 27, 28, 34] or proposing new energy-based algorithms [19] to enhance robustness, realism, and generalizability to other flexible materials. As hardware performance has evolved, other work has proposed to exploit the parallel computing power of modern GPUs [33, 40]. These approaches achieve a high degree of realism at the cost of high computational costs. Therefore, physics-based simulator is unsuitable when real-time performance is required, or computational power is limited. On the other hand, linear blend skinning is the standard approach used in computer graphics for 3D model animation [14, 15, 18, 22, 36, 37] applications where performance is a priority. Each vertex of the object being animated is connected to the skeleton by a set of hybrid weights that are used for linear combinatorial joint transformations. In the field of garment animation, garments are attached to a skeleton that drives body movement. This approach has also been extensively studied.

2.2. Learning-Based Methods

Recently, learning-based approaches have been proposed to deal with the motion of garments. Researchers use learning-based approaches to obtain pose space deformations and linear skinning to generate garment animations. Lähner et al. [17] also proposed nonlinear mapping by combining temporal features processed by RNN for linearly learning pose space deformations of garments. Later, Santesteban et al. [31] proposed an explicit MLP-based nonlinear mapping method for spatial pose morphing single template garments. The main drawback of these approaches is that the pose space deformation must be learned for each template garment, which requires new simulations to obtain the corresponding data. To address this problem, many researchers have proposed an extended human body model (SMPL [20]) that encodes garments as additional displacements and topologies as a subset of vertices [1, 2, 5, 8, 26]. Alldieck et al. [1, 2] proposed a single human body and garment model, first as vertex displacements and then as texture replacement mappings, to infer 3D shapes from a single RGB image. Similarly, Bhatnagar et al. [8] learn the space of body deformations to encode garments, an additional segmentation to separate body and garment, and infer 3D garments from RGB. Jiang et al. [13] propose a approach to retrieve 3D garments from images and predict the corresponding mixture weights. The weights of recent masked vertices were used as the SMPL skeleton for the markers. Patel et al. [26] encoded several garment types as a subset of body vertices. They propose a strategy to handle fabric details associated with high-frequency locations for different body types and garment styles. Bertiche et al. [5] encode thousands of garment types to the top of the body by masking the body vertices. They learned a continuous space of garment types on which they subsequently adjusted vertex deformations with the pose. Using a human body model to represent garments allows one model to handle multiple types.

However, all of these approaches take the parameters of the linear transformation model from the data and eventually use a linear stripping method to obtain the results. Since the garment motion is highly nonlinear, using a linear model to solve for it can significantly reduce the realism of the results. These approaches have another drawback. Most of them directly encode the garment as one with the same topology as the body, which greatly limits the usefulness and scalability of the model. We propose to encode garments into latent space and generate garment animations directly using a completely neural network-based approach, which will maintain the high performance and stability of the model and extend the garment types to garments with different topologies from the human body.

In contrast to previous approaches that encode homological garments with body homology, the recent DeePSD [7] extends garment types to multiple topologies. This dramatically improves the applicability and expandability of this domain. We follow a similar basic idea but propose a fully neural network-based model for learning. In addition, our approach considers temporal dependence and can handle animated models that produce high realism and more delicate wrinkles.

3. Methodology

In this section, we describe specifically how our GSNet generates garment animations from human motions, and Figure 2 shows a general overview of our approach. Next, we overview each module in our model in individual chapters.

3.1. Garment Model

Similar to state-of-the-art approaches of learning-based [5, 10, 26, 31, 35], we exploit and extend existing human body models [9, 20]. More specifically, we construct our human body representation based on the popular SMPL [20] human body model. SMPL [20] encodes the body by deforming the manipulated human template based on shape β and pose θ_t related deformations learned from the data. Then, we define our garment model as:

$$G_t = W\left(\mathbf{T}, G_{t-1}, S\left(\beta, \theta_t\right)\right) \tag{1}$$

where $G_t \in \mathbb{R}^{N\times 3}$ is the predicted deformed target garment mesh and N is the number of garment template vertices, $G_{t-1} \in \mathbb{R}^{N\times 3}$ is the garment vertex attribute of the previous moment and $\mathbf{T} \in \mathbb{R}^{N\times 3}$ is the garment outfit, $S(\cdot)$ is used to obtain information on key joints of human body

with body shape $\beta \in \mathbb{R}^{10}$ and body motion $\theta_t \in \mathbb{R}^{72}$ from SMPL [20] model. Our goal is to train $W(\cdot)$ as defined in Eq 1.

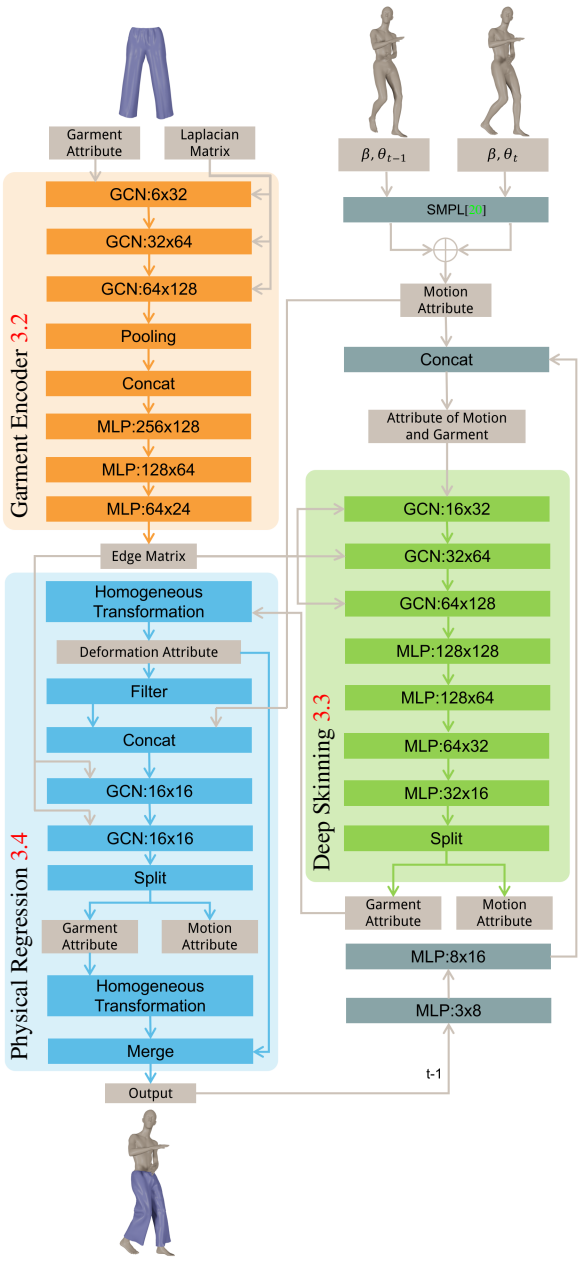


Figure 2. An overview of our approach, we use the Garment Encoder (Section 3.2) to process garment with multiple topologies. Then, we use Deep Skinning (Section 3.3) to aggregate body motion information to garment attributes to get garment deformation attributes. Finally, the Physical Regression (Section 3.4) helps to correct the vertices that do not conform to the physical constraints. The final result is obtained.

3.2. Garment Encoder

The goal of the garment encoder is to handle garments of multiple topologies. The dominant representation in garment animation uses a mesh model, which is very similar to the graph structure. Therefore, we choose graph convolution [38] to handle problems related to garment animation. We can easily obtain vertex properties and edge information in the mesh model. When extracting the garment features from the garment encoder, we preprocess the edge information into a normalized Laplace matrix in order to make the aggregated features more accurate. The normalized Laplacian matrix is generated by the following formula:

$$\mathrm{L_n} = \mathrm{D}^{-\frac{1}{2}}\mathrm{L}\mathrm{D}^{-\frac{1}{2}} \tag{2}$$

where D denotes the degree matrix, and L denotes the standard Laplacian matrix. Specifically, the value of each element can be expressed as:

$$\mathrm{L_{ij}} = \begin{cases} 1, & \text{if i = j} \\ \frac{-1}{\sqrt{\mathrm{D_{ii}}}\sqrt{\mathrm{D_{jj}}}}, & \text{if i, j in edge set} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The initial information $\in \mathbb{R}^{N\times 6}$ of the garment vertex contains its location in the world space, material, and relaxation schedule as properties. We take this information and pass it to the graph convolution layer for processing, the whole graph convolution formula can be expressed as:

$$X_{l+1} = \sigma\left(\mathrm{L_n} X_l W_{(l)} + b_l\right) \tag{4}$$

where $\mathrm{L_n}$ is normalized Laplacian matrix, X_{l+1} and X_l are are the feature matrices of the layer $l+1$ and l, W_l is a trainable weight matrix for layer l, b_l is a trainable bias for layer l.

After three layers of graph convolution processing, we get the features$\in \mathbb{R}^{N\times 128}$ of each vertex. Then, to make the features more visible, we pass the result to the maximum pooling layer to get a global feature$\in \mathbb{R}^{1\times 128}$. Then, we splice the global feature to each vertex feature to get a result$\in \mathbb{R}^{N\times 256}$ that represents the feature of garment in latent space.A general overview of the clothing encoder is shown in orange in Figure 2

3.3. Deep Skinning

Deep skinning aims to aggregate changes in body motion onto a garment and obtain information about the changes corresponding to it. Similarly, we still choose graph convolution for information aggregation. The most popular body motion representation model is SMPL [20], which uses 24 vital skeletal joints to represent body motion features. In this paper, we choose SMPL [20] as our body model, which only needs to pass shape β and pose θ_t parameters to obtain the motion information of vital skeletal joints$\in \mathbb{R}^{24\times 16}$

at time t. After obtaining the body motion skeletal joints, we take the garment attributes$\in \mathbb{R}^{N\times 3}$ of the previous moments $t-1$ and pass them to the two fully connected layers to get the result$\in \mathbb{R}^{N\times 16}$ with the same dimension as the skeletal joints, then we construct the set $\in \mathbb{R}^{(N+24)\times 16}$ of vertices in the graph convolution from the garment vertices at the moment $t-1$ together with the body motion skeletal joints.

In addition, we need to obtain the edge information between the garment vertices and the vital skeletal joints. Since the garment vertices are not directly and explicitly connected to the skeletal joints, the edge matrix cannot be generated as a conventional graph convolution like the normalized Laplacian matrix. For this reason, we use neural networks to obtain the edge information between garment vertices and skeletal joints. The specific implementation is to take the result$\in \mathbb{R}^{N\times 256}$ of the garment encoder and pass it through three fully connected layers to obtain the output$\in \mathbb{R}^{N\times 24}$ as the edge information between the garment vertices and the skeletal joints of the body. Then this result is transformed into a sparse matrix$\in \mathbb{R}^{(N+24)\times(N+24)}$, which is used as the edge information of the graph formed by the body skeleton joints and the garment vertices.

After obtaining the vertex set$\in \mathbb{R}^{(N+24)\times 16}$ and the sparse matrix$\in \mathbb{R}^{(N+24)\times(N+24)}$ of edges, we construct a graph structure from them and pass them to the graph convolutional neural network. Similarly, the convolutional network is structured as in Eq 4. After three layers of graph convolution, we aggregate the body motion information to the garment vertices. Then we split the graph to obtain the set of vertices$\in \mathbb{R}^{N\times 128}$ to the garment and pass it to the fully connected layer. After four fully connected layers, the final result$\in \mathbb{R}^{N\times 16}$ is the deformation attributes of each vertex of the garment. We take the deformation attributes and use the homogeneous transformation to generate the result$\in \mathbb{R}^{N\times 3}$ as the vertex at the moment t of the garment.

3.4. Physical Regression

After deep skinning, we get the current deformation attributes of the garment, but not all of them conform to the physical constraints. Therefore, we also set up a physical regressor to correct those garment attributes that do not conform to the physical constraints.

We designed a filter to obtain the garment vertices that do not conform to the physical constraint with the filter condition: $\mathbf{d}_{j,i}\cdot\mathbf{n}_j < 0$ where $\mathbf{d}_{j,i}$ is the vector going from the j-th vertex of the body to the i-th vertex of the garment, $\mathbf{n}_j$ is the j-th vertex normal of the body. The result of the two vectors is less than 0, which means that the angle between them is greater than $90°$, and there is a situation that does not satisfy the physical constraint. We form a graph structure of these vertices with skeletal points and then use graph convolution combined with unsupervised learning to obtain the correct vertices that conform to the physical constraint. The specific process is similar to deep skinning. Finally, we fuse the corrected vertices with the original ones to obtain the final result $G_t^{N\times 3}$. This result $G_t^{N\times 3}$ is the garment deformation corresponding to current body motion(β,θ_t).

3.5. Loss Function

The loss function is a key component of our learning-based algorithm. We use different loss terms to train our approach to get realistic results that conform to physical constraints.

L2 Loss : The goal of this term is to minimize the Euclidean error with the data based on physical simulations. This allows the predicted results to conform as closely as possible to the original similar shape.The L2 loss on the positions can be expressed as:

$$\mathcal{L}_{data} = \sum \|\mathbf{V}_{pred} - \mathbf{V}_{pbs}\|^2 \tag{5}$$

where V_{pred} is the position on the predicted garment vertices, V_{pbs} is the position on the physical simulation-based vertices.

Physical Loss : L2 loss only ensures that the output results remain similar to the shape of the garment template. We also need to define some physical constraints to make the output results more visually appealing. Meanwhile, in order to keep a fair comparison with other approaches, we use the following three loss terms.:

$$\mathcal{L}_{edge} = \sum_{e\in E} \|e - e_{\mathbf{T}}\|^2 \tag{6}$$

where E is the set of edges of the given garment template, e is the predicted edge length and $e_{\mathbf{T}}$ is the edge length on the garment template $\mathbf{T}$. $\mathcal{L}_{edge}$ enforces the output to have the same edge lengths as the input template. Then, in order to yielding locally smooth surfaces, we define a blend loss $\mathcal{L}_{blend}$ as :

$$\mathcal{L}_{blend} = \lambda_B \Delta(\mathbf{n})^2 \tag{7}$$

where $\Delta(\cdot)$ is the Laplace-Beltrami operator applied to vertex normals $\mathbf{n}$ of the predicted output. λ_B is to avoid excessive flattening. To handle collisions against the body, we define a collision loss $\mathcal{L}_{collision}$ as:

$$\mathcal{L}_{\text{collision}} = \sum_{(i,j)\in A} \min\left(\mathbf{d}_{j,i}\cdot\mathbf{n}_j - \epsilon, 0\right)^2 \tag{8}$$

where A is the set of correspondences (i,j) between predicted output and body 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 garment, $\mathbf{n}_j$ is the j-th vertex normal of the

body and ϵ is a small positive threshold to increase robustness. This loss is a simplified formulation that assumes garment is close to the skin, and penalizes outfit vertices placed inside the skin.Thus, the whole physical loss $\mathcal{L}_{physical}$ is defined as:

$$\mathcal{L}_{\text{physical}} = \mathcal{L}_{\text{edge}} + \mathcal{L}_{\text{blend}} + \mathcal{L}_{\text{collision}} \tag{9}$$

which can guide our model to obtain realistic results.

4. Experiments

This section describes our implementation and shows our results in several complex benchmarks. We not only perform complex comparisons with current state-of-the-art learning-based approaches but also with the performance of popular physics-based simulation approach.

4.1. Datasets and Experimental Setup

From all the current public datasets, only CLOTH3D [5] contains enough garment variability to implement our approach. It contains about 7.5K sequences, each with a different template in the resting pose. These costumes are simulated on top of a 3D animated person (SMPL) [20], each with a different body type. Similarly, we use the SMPL [20] framework to drive the human and garment motion of our model. Next, we used the same partitioning strategy as DeePSD [7] for the dataset to enable a fair comparison, and we subsampled 50k training frames and 5k test frames. There was no overlap between the training frames and the test frames. Models. Each model was trained for 20 epochs with 12500 steps per epoch using the adam [16] optimizer with a batch size of 4.

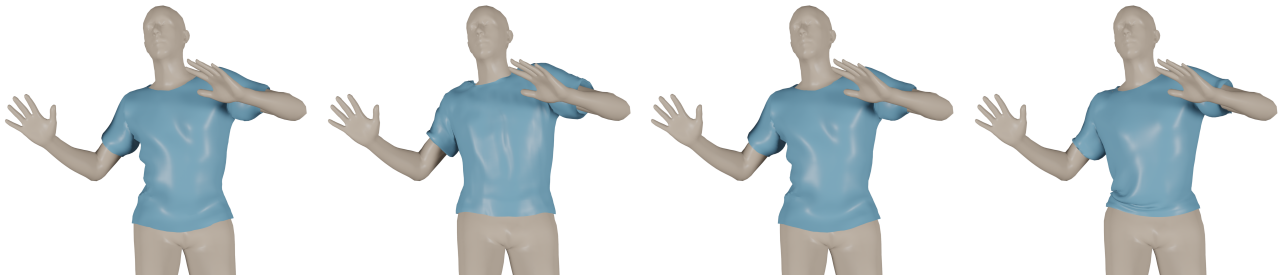
4.2. Qualitative Results

We compared our approach qualitatively with DeePSD [7], PBNS [6], and TailorNet [26].As seen in Figure 3, our approach achieves better visual results on garments with simple topologies when compared to state-of-the-art approaches. More specifically, we also compare in detail with each approach individually. In comparison with PBNS [6], our approach has good results not only on garments with the same topology as the body but also on garments with a different topology from the body, and our approach has realistic visual effects. This is due to the garment encoder in our approach, which abstractly extracts features from garments with different topologies and encodes them into the latent space, which gives the model the ability to handle garments with different topologies. PBNS [6], on the other hand, simply encodes all garments to be consistent with the body structure, resulting in a minimal expansion capability. The experimental results for different topologies are shown in Figure 4. As shown in Figure 5, compared to DeePSD [7] and TailorNet [26], for the same garment template, without physical constraints or post-processing, the penetration rate of our approach is shallower because we include temporal dependence in the network, which allows the model to obtain more contextual information. Temporal dependence in the network leads to more accurate results and lower penetration. As shown in Figure 6, in comparison with DeePSD [7], the effect generated by our approach is visually more continuous and vivid in the complete animation sequence due to the temporal dependence added to our network. In contrast, the overall animation effect of DeePSD [7] is very stiff because it only generates the frame based on the action without considering the temporal context. More specific results can be observed in our supplementary material.

4.3. Quantitative Results

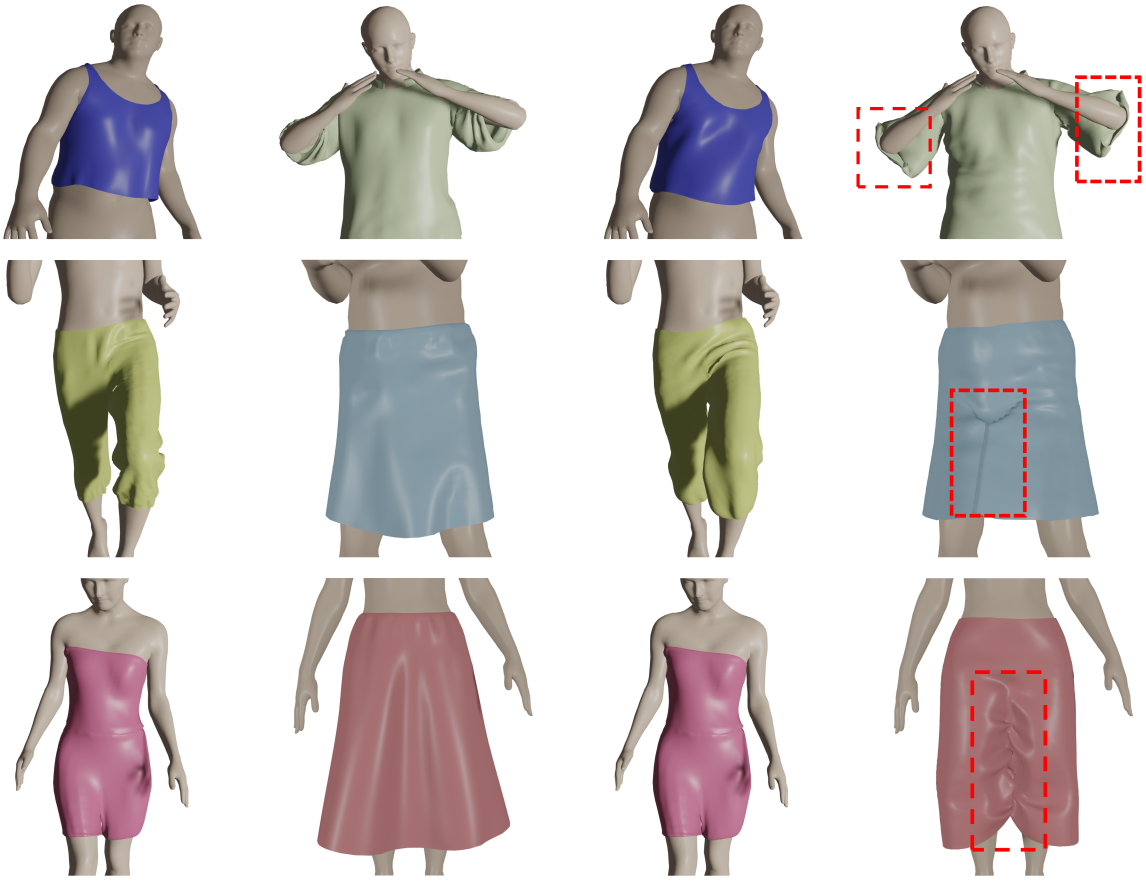
In addition to the qualitative experiment, we also conducted detailed quantitative experiment. In order to allow for a fair comparison with the state-of-the-art approaches, we used the same metrics as the state-of-the-art approaches. Euclidean error, edge elongation/compression, bending angle between vertex normals, and collision rate between garment and body vertices. The lower the Euclidean error, the closer the results will be to ground truth. The three metrics of edge elongation/compression, bending angle between vertex normals, and collision rate between garment vertices and body vertices were used to determine if the results met the physical constraints. The lower these three metrics are, the more the results conform to the physical constraints.

We tested the state-of-the-art approaches separately on the same dataset, and the results are shown in Table 1, where we can see that the Euclidean error of our approach is significantly lower than that of the other approaches. our approach also achieves good results in the other three metrics. In Table 1, we observe that the Euclidean error of PBNS [6] is much higher than all other approaches, so we perform further experiment. We divide the dataset into six categories with different topologies according to the garment topology. They are Tshirt, Top, Trouser, Jumpsuit, Skirt, and Dress. Detailed comparison results are shown in Table 2, and we can find that our approach achieves excellent results on garments with different topologies. Further, we can find that PBNS [6] produces a more significant Euclidean error when dealing with two types of garments, Skirt and Dress, because most of the topologies of these garments are not the same as the body topology. PBNS [6] encodes them to the body for processing, leading to a significant error. Moreover, we can also observe from either Table 1 or Table 2 that our approach also has a good advantage over other approaches in all metrics. This is because our approaches is entirely nonlinear, making the model fitting ability more robust and reducing error. The rest of the approaches use the network to solve for the skinning weights and then combine



(a) PBNS[6] (b) DeePSD[7] (c) TailorNet[26] (d) GSNet(Ours)

Figure 3. Compared to state-of-the-art approaches such as PBNS [6], DeePSD [7], and TailorNet [26], our approach achieves realistic results on simple topologies such as T-shirts.



(a) GSNet(Ours) (b) PBNS[6]

Figure 4. GSNet(Ours) designs a garment encoder that encodes garments with different topologies into the latent space, enhancing the expansion capability. GSNet(Ours) can handle garments with different topologies, as shown in the left column of each subfigure. GSNet(Ours) can also handle garments with different topologies from the body, as shown in the right column of each subfigure. PBNS [6] directly encodes the garment into a structure consistent with the body so that an unreasonable effect can occur.

them with a linear skinning function to obtain the results, which leads to a less robust ability to fitting.

In addition to the above metrics for quantitative experiment, we also tested the stability and performance of the model. Figure 7 shows the comparison results with state-of-the-art approaches, from which we can observe that the gap between our approach is more stable on the training and test sets. Through further experiment, we found that DeePSD [7] training results become worse as the epoch increases. At the same time, our approach can maintain sta-

Figure 5. GSNet(Ours) introduces temporal dependency so that the model can obtain more temporal context information and the prediction results are more and more accurate. State-of-the-art approaches such as TailorNet [26] and DeePSD [7] are based on frame-by-frame actions to predict, leading to limited information acquisition by the model. The generated results could be more realistic and rely heavily on post-processing to repair results.

(a) DeePSD[7] (b) Simulation (c) GSNet(Ours)

Figure 6. GSNet(Ours) introduces temporal dependency, which help to generate realistic effects as in physical simulations. DeePSD [7] will have a more rigid effect because it is based on static method.

ble results, and the related data are shown in Figure 8. In terms of performance experiment, we compared it with the most popular physical simulator [24]. Our computational performance is much better than the physical simulator under the same costume template, and related result data are organized. The relevant result data are organized in Figure 9.

Table 1. GSNet(Ours) achieves better results in all four evaluation metrics than state-of-the-art approaches. The most obvious advantage is the Euclidean error evaluation. This is because GSNet(Ours) processing is based on nonlinearity, which allows for a more robust fitting of our model and demonstrates an advantage in the experiment.

	Error	Edge	Bend	Collision
PBNS[6]	64.51	0.72	0.033	0.96%
DeePSD[7]	29.75	1.13	**0.029**	1.29%
TailorNet[26]	21.63	0.68	0.031	0.81%
GSNet(Ours)	**13.15**	**0.60**	0.037	**0.7%**

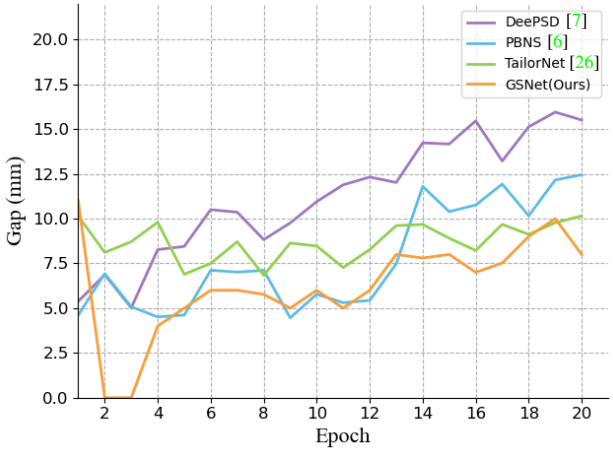


Figure 7. Comparison of the gap between the training and test sets, our model has a much smaller gap between the training and test sets.

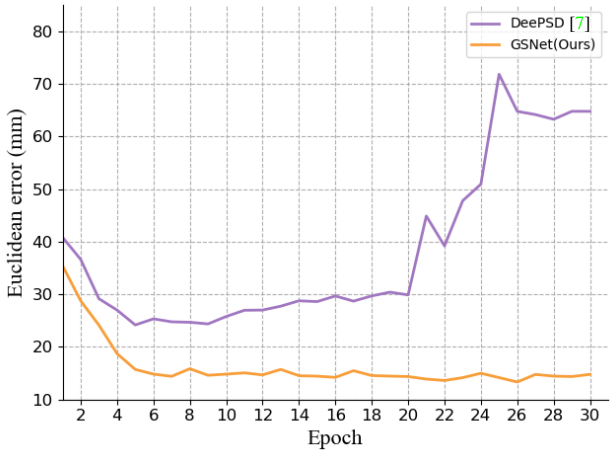


Figure 8. Comparison of stability, our approach is less susceptible to the influence of epoch hyperparameters and more stable.

Table 2. We divide the dataset into six categories according to different garment topologies and test these six templates. GSNet(Ours) performs well on garments with different topologies, demonstrating good stability and expandability.

	Method	Error	Edge	Bend	Collision
Tshirt	DeePSD[7]	27.73	0.97	**0.031**	0.91%
	PBNS[6]	16.71	**0.68**	0.037	**0.65%**
	TailorNet[26]	18.43	0.81	0.045	0.87%
	GSNet(Ours)	**13.28**	0.70	0.040	0.74%
Top	DeePSD[7]	27.24	0.71	0.026	1.02%
	PBNS[6]	14.64	0.43	**0.017**	**0.51%**
	TailorNet[26]	14.22	**0.36**	0.041	0.67%
	GSNet(Ours)	**12.72**	0.54	0.028	0.85%
Trousers	DeePSD[7]	23.35	1.12	0.027	1.27%
	PBNS[6]	14.11	**0.53**	**0.018**	**0.41%**
	TailorNet[26]	13.62	0.59	0.029	0.58%
	GSNet(Ours)	**12.80**	0.63	0.033	0.83%
Jumpsuit	DeePSD[7]	25.04	1.17	0.035	1.36%
	PBNS[6]	18.31	**0.24**	**0.007**	**0.68%**
	TailorNet[26]	17.19	0.61	0.019	0.73%
	GSNet(Ours)	**12.86**	0.56	0.036	0.82%
Skirt	DeePSD[7]	42.48	1.38	**0.038**	1.26%
	PBNS[6]	98.27	1.17	0.076	1.01%
	TailorNet[26]	17.72	**0.53**	0.041	0.84%
	GSNet(Ours)	**13.85**	0.66	0.048	**0.66%**
Dress	DeePSD[7]	44.45	1.44	**0.042**	1.24%
	PBNS[6]	50.69	0.85	0.045	1.64%
	TailorNet[26]	27.67	0.72	0.058	1.14%
	GSNet(Ours)	**13.61**	**0.58**	0.044	**0.54%**

Table 3. Results of the architecture ablation experiment.The Euclidean error becomes even lower after adding the temporal dependency.

	Euclidean error (mm)
Baseline	20.92
+Garment Temporal Dependency	15.51
+Pose Temporal Dependency	**14.21**

4.4. Ablation Experiment

In order to determine how much different processes contribute to the outcomes, we conducted ablation experiment.

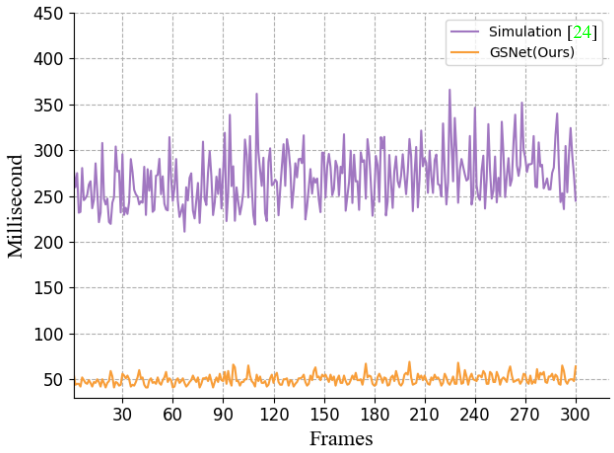


Figure 9. In a performance comparison, our approach is six to seven times more efficient than physical simulation in processing time per frame using the garment template in Figure 3.

Table 3 shows our results. The majority of other techniques are static in nature and do not account for the motion factor of the previous instant. We discovered through ablation experiment that the effect is enhanced by embedding human motion dependence in addition to embedding garment motion dependence, which has the largest effect. This implies that the animation that is generated is significantly influenced by temporal dependence. The output of the experiment shows that, in addition to the motion, only the effect of time needs to be taken into account when analyzing the effects of garment motion.

5. Conclusions, Limitations, and Future Work

In this work, we propose a new approach to garment animation, whereas the past approaches relied heavily on linear skinning. While our approach is completely based on learning, it eliminates linear skinning. The efficiency is improved, and it uses for a garment with multiple topologies. We also introduce a temporal dependency in the model, using the previous results as input for the current moment. In summary, we have developed a approach that can solve the animation of garment with multiple topologies, and our performance is ahead of other current approaches.

At the same time, we are aware of our limitations. First, our treatment of the time dependency is too rudimentary; we take the previous moment's output as the next moment's input directly. For stability, a temporal neural network should be introduced to compute it. Second, although it does not affect the visual effect, our approach is not obvious enough in generating geometric details (wrinkles), and we also observe how recent studies model fine geometric details (wrinkles) based on complexity. We believe that the best way to deal with garment folds is through normal mapping generation, and current work in this area seems promising [17, 41]. We set this up as future work.

Acknowledgement

This work is partly supported by Educational Commission of Hubei Province of China (NO.D20211701) and Cixi Science and Technology Bureau (NO.2021Z069) and National Natural Science Foundation of China, and Engineering Research Center of Hubei Province for Clothing Information.

References

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

[2] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2293–2303, 2019. 3

[3] A. Arsalan Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum. 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*, pages 1511–1519, 2017. 2

[4] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998. 2, 3

[5] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 3, 6

[6] H. Bertiche, M. Madadi, and S. Escalera. Pbns: physically based neural simulation for unsupervised garment pose space deformation. *ACM Transactions on Graphics (TOG)*, 40(6):1–14, 2021. 2, 6, 7, 9

[7] H. Bertiche, M. Madadi, E. Tylson, and S. 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. 2, 3, 6, 7, 8, 9

[8] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 5420–5430, 2019. 3

[9] A. Feng, D. Casas, and A. Shapiro. Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, pages 57–64, 2015. 3

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

[11] X. Han, C. Gao, and Y. Yu. 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] S. Hong and H. Kim. An analytical model for a gpu architecture with memory-level and thread-level parallelism awareness. In *Proceedings of the 36th annual international symposium on Computer architecture*, pages 152–163, 2009. 2

[13] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao. Bcnet: Learning body and cloth shape from a single image. In *European Conference on Computer Vision*. Springer, 2020. 3

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

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

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] Z. Lahner, D. Cremers, and T. Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *Proceedings of the European conference on computer vision (ECCV)*, pages 667–684, 2018. 3, 10

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

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

[20] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2, 3, 4, 6

[21] M. Madadi, H. Bertiche, and S. Escalera. Smplr: Deep learning based smpl reverse for 3d human pose and shape recovery. *Pattern Recognition*, 106:107472, 2020. 2

[22] N. Magnenat-Thalmann, R. Laperrire, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *In Proceedings on Graphics interface'88*. Citeseer, 1988. 2, 3

[23] D. Merrill and A. Grimshaw. High performance and scalable radix sorting: A case study of implementing dynamic parallelism for gpu computing. *Parallel Processing Letters*, 21(02):245–272, 2011. 2

[24] R. Narain, A. Samii, and J. F. O'brien. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)*, 31(6):1–10, 2012. 2, 3, 8, 10

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

[26] C. Patel, Z. Liao, and G. 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. 2, 3, 6, 7, 8, 9

[27] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation'97*, pages 177–189. Springer, 1997. 2, 3

[28] X. Provot et al. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147. Canadian Information Processing Society, 1995. 2, 3

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

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

[31] I. Santesteban, M. A. Otaduy, and D. Casas. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, volume 38, pages 355–366. Wiley Online Library, 2019. 2, 3

[32] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Ng. Convolutional-recursive deep learning for 3d object classification. *Advances in neural information processing systems*, 25, 2012. 2

[33] M. Tang, R. Tong, R. Narain, C. Meng, and D. Manocha. A gpu-based streaming algorithm for high-resolution cloth simulation. In *Computer Graphics Forum*, volume 32, pages 21–30. Wiley Online Library, 2013. 2, 3

[34] T. Vassilev, B. Spanlang, and Y. Chrysanthou. Fast cloth animation on walking avatars. In *Computer Graphics Forum*, volume 20, pages 260–267. Wiley Online Library, 2001. 2, 3

[35] R. Vidaurre, I. Santesteban, E. Garces, and D. Casas. Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum*, volume 39, pages 145–156. Wiley Online Library, 2020. 3

[36] R. Y. Wang, K. Pulli, and J. Popović. Real-time enveloping with rotational regression. In *ACM SIGGRAPH 2007 papers*, pages 73–es. 2007. 2, 3

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

[38] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019. 4

[39] N. Wu, Q. Chao, Y. Chen, W. Xu, C. Liu, D. Manocha, W. Sun, Y. Han, X. Yao, and X. Jin. Example-based real-time clothing synthesis for virtual agents. *arXiv preprint arXiv:2101.03088*, 2021. 2

[40] C. Zeller. Cloth simulation on the gpu. In *ACM SIGGRAPH 2005 Sketches*, pages 39–es. 2005. 2, 3

[41] M. Zhang, T. Wang, D. Ceylan, and N. J. Mitra. Deep detail enhancement for any garment. In *Computer Graphics Forum*, volume 40, pages 399–411. Wiley Online Library, 2021. 10