




# Detail-Aware Deep Clothing Animations Infused with Multi-Source Attributes

T. Li,<sup>1</sup>  R. Shi<sup>2</sup> and T. Kanai<sup>2</sup>

<sup>1</sup>Faculty of Information Technology, Beijing University of Technology, Beijing, China

<sup>2</sup>Graduate School of Arts and Sciences, The University of Tokyo, Tokyo, Japan  
rui-shi@outlook.com, kanait@acm.org

---

## Abstract

*This paper presents a novel learning-based clothing deformation method to generate rich and reasonable detailed deformations for garments worn by bodies of various shapes in various animations. In contrast to existing learning-based methods, which require numerous trained models for different garment topologies or poses and are unable to easily realize rich details, we use a unified framework to produce high fidelity deformations efficiently and easily. Specifically, we first found that the fit between the garment and the body has an important impact on the degree of folds. We then designed an attribute parser to generate detail-aware encodings and infused them into the graph neural network, therefore enhancing the discrimination of details under diverse attributes. Furthermore, to achieve better convergence and avoid overly smooth deformations, we proposed to reconstruct output to mitigate the complexity of the learning task. Experimental results show that our proposed deformation method achieves better performance over existing methods in terms of generalization ability and quality of details.*

**Keywords:** cloth animation, animation systems, animation

**CCS Concepts:** • Computing methodologies → Neural networks; Animation

---

## 1. Introduction

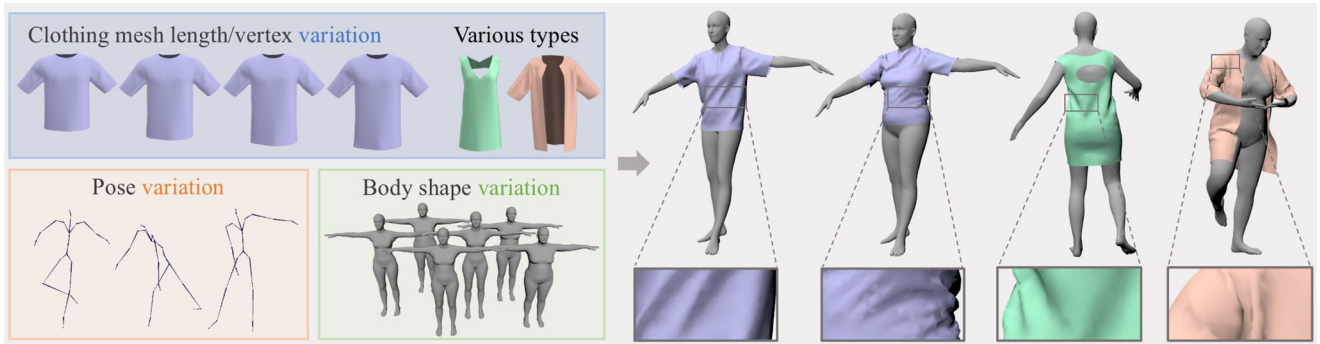
Clothing animation is a fundamental topic in computer graphics, aiming to generate realistic clothing deformation effects for many applications including virtual try-on, video games and films. With the progress of the graphics field, users are paying more attention to the visual effects of garments, including how they interact more realistically with the body and how wrinkles increase or decrease with different movements. High-quality clothing deformations provide users with convenience during online shopping or provide an immersive experience for entertainment.

To meet the needs of producing high-quality clothing animations, predominant approaches are based on physics-based simulation [NSO12, NMK\*06]. Despite the convincing effects provided by these methods, deployment to real-time applications is still challenging due to the high costs of computer simulation process. [supplementary material](#)

To overcome high computational costs and simplify the deformation process, learning-based solutions [dASTH10] are proposed to approximate clothing deformations according to relevant influ-

encing factors (*e.g.*, motion and shape of the body). While these methods can roughly imitate the behaviour of clothing animation, there still remain issues in terms of generalization and quality of details.

Most state-of-art learning-based studies [SOC19, PLPM20, TBTP20] adopt multi-layer-perceptron (MLP) models to predict the nonlinear deformations of garments. Although the predicted results contain plausible wrinkles, the trained model cannot generalize to new garments because both input and output are vectors of restricted size (usually related to the number of vertices), resulting in the training and test targets being forced to have the same number of vertices. Even though a constant mesh topology with style parameters [PLPM20] can cope with different garments, it cannot represent fine details with reasonable mesh resolution when the deformation targets are highly variable (*e.g.*, extremely long and short garments, or t-shirt and jacket). Furthermore, because of the limited ability of MLPs to understand 3D information, a great number of parameters is usually required to realize the deformation approximation for specific mesh topologies. On the other hand, solutions based on graph neural networks [CMM\*20, GCP\*22] can effectively address



**Figure 1:** We present a novel learning-based method for automatically generating detail-aware deformations for diverse garments worn by different body shapes in arbitrary poses. Through our method, the model can easily make reasonable approximations for individualized deformations caused by different attributes.

the generalization limitation of MLPs, as their input and output are 3D mesh features and the trained parameters are independent of the mesh topology and the number of vertices. However, the approximated garments tend to be overly smooth and lack rich wrinkles [GCS\*19]. To enable realistic clothing deformations, existing graph learning-based research [VSGC20] has to trade pose-variation for realism, which only predicts the deformation in t-pose.

The main reason why learning-based methods for clothing animation need to weigh the above aspects is: the extreme complexity of the fine deformation prediction of garments in multiple states (under various postures, worn by various bodies, *etc.*). Our method essentially overcomes this ‘complexity’ and uses one framework to efficiently generate high-quality deformations with fine details (see Figure 1). Deformations can be approximated in two steps: (1) learn a model to globally drape the garment on the target body in a certain pose, (2) learn an additional model to produce the high-frequency wrinkles based on the corresponding coarse deformation. The overview of the method is shown in Figure 2. Specifically, our technical contributions are three-fold:

- To account for complicated and irregular detailed wrinkles, we first discuss that the fit between the garment and body influences the degree of wrinkles: loose clothes have smoother, sparser and wide wrinkles, while tight clothes have thinner, denser and narrow wrinkles. Therefore, we parametrize the relationship and propose the fit parameter, which is regarded as one of the attributes.
- To make the model generalized and effectively map relevant influencing attributes (*i.e.*, fit, body shape and pose) to deformation details, we design an attribute parser to generate detail-aware encodings and then infuse them into the graph neural network. This infusion maps the original graph features to representative features that are adaptive to the corresponding attributes, providing a meaningful signal to the model and learning realistic deformations in a detail-aware manner.
- To facilitate the deformation learning and achieve high-quality predictions, we address complexity fundamentally from the novel perspective of output reconstruction. Existing studies always directly output the three-dimensional vector (position or displacement) of each vertex where the value of each dimension ranges from negative infinity to positive infinity, which makes it difficult

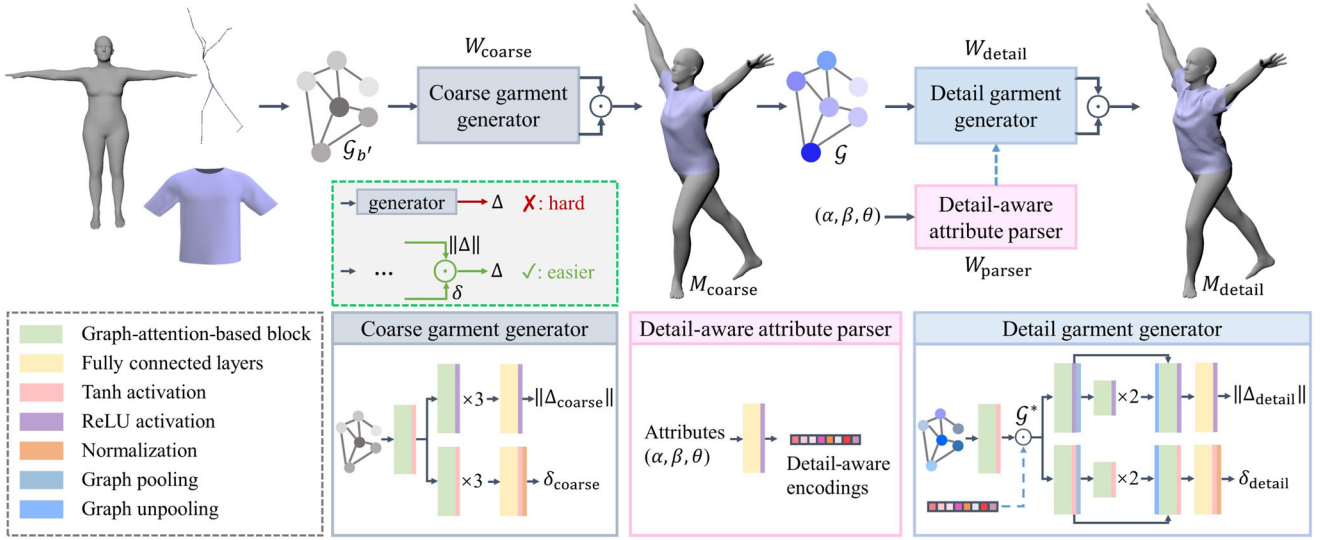
for the training to converge to a reasonable range and the prediction results tend to be overly smooth. To address this problem, we decompose the output vector as the combination of magnitude and direction where the value range of the magnitude is greater than zero and the value range of the direction is from  $-1$  to  $1$ . This strategy plays a crucial role in the learning of fine deformations, since it greatly reduces the range of output variables, thereby mitigating the complexity of the task.

To the best of our knowledge, our study has been the first to enable unified models to realize detail-aware deformations for garments with various mesh geometries worn by diverse body shapes in any posture. Our experiments confirm that our proposed method outperforms existing clothing animation methods in terms of generalization and deformation quality.

## 2. Related Work

In this section, we first discuss existing clothing animation methods by classifying them into physics-based simulation and learning-based models. Then, we also introduce the latest investigations on learning-based deformation.

**Physics-based simulation.** Pioneering studies achieve realistic clothing animations based on geometric constraints [LC04, SSIF09, RPC\*10], however, they always suffer from instability and high computational cost. In order to make the simulation efficient, research in Müller and Chentanez [MC10] computes wrinkles by a static solver and adds them on the coarse base mesh. As a similar idea on adding fine details on low-quality cloth, Gillette *et al.* [GPV\*15] propose tracing wrinkle paths on the coarse mesh following the per-triangle compression field. To accelerate the computation, recent researchers are also making efforts to improve GPU-based algorithms. For example, yarn-level contact can be modelled implicitly with GPU in [CLMMO14]. Ni *et al.* [NKT15] present an algorithm to simulate cloth with complex collisions using a parallel run-time system. To exploit high parallel performance, a matrix assembly algorithm is proposed [TWT\*16] which can accurately solve the linear system. For clothes with more than 50,000 vertices, research in Wu *et al.* [WWYW20] can still achieve fast simulations because of the effective conversion of continuous constraints. In



**Figure 2:** Overview of proposed method pipeline. Given a garment with an arbitrary mesh topology, a target body with any shape, and a random animated posture, our method is able to approximate high-quality clothing deformation with expressive detail wrinkles. Our key contribution is to address the challenge of ‘complexity’ by designing a two-step framework with ideas of proposing the fit parameter  $\alpha$ , detail-aware attribute parser and output reconstruction (from the displacement vector  $\Delta$  to its magnitude  $\|\Delta\|$  and direction  $\delta$ ). First, the constructed graph  $\mathcal{G}_b'$  is fed into a coarse garment generator  $W_{\text{coarse}}$  to predict the decomposed components  $\|\Delta_{\text{coarse}}\|$  and  $\delta_{\text{coarse}}$  of the coarse corrective displacement before realizing coarse deformation prediction  $M_{\text{coarse}}$ . Next, we build a graph  $\mathcal{G}$  based on the generated deformation  $M_{\text{coarse}}$ . Instead of directly applying attributes to each graph node, we further propose a detail-aware attribute parser  $W_{\text{parser}}$  to generate detail-aware encodings and infuse them into the original graph to obtain the representative  $\mathcal{G}^*$ . Then, a detail garment generator is designed to process features of  $\mathcal{G}^*$  and output  $\|\Delta_{\text{detail}}\|$  and  $\delta_{\text{detail}}$  in each branch. Two predictions are finally multiplied and added to the  $M_{\text{coarse}}$  to realize the ultimate detail clothing deformation  $M_{\text{detail}}$ .

practice, we usually use physics-based simulations as ground truth data for learning-based deformation and train the model to be able to estimate effects close to those of the physics-based simulations.

**Learning-based clothing models.** Inspired by the success of deep learning, a number of works are attempting to learn the deformation as a function of relevant parameters, where relevant parameters include closest body vertex position, associate body skinning weights, joint rotation angle, *etc.*

To resolve the high computational costs of physics-based simulation while realizing non-linear clothing behaviours, Santesteban *et al.* [SOC19] propose a two-level strategy to generate clothing deformations, where the first step is to use MLPs to learn the global fit and the second step is to use recurrent neural networks to learn the wrinkles. Also in order to estimate cloth deformations with fine details, TailorNet [PLPM20] adopts multiple MLPs to realize the task, in which low-frequency deformations are predicted using a simple MLP model, and high-frequency deformations are predicted using the mixture of multiple MLPs. To model how people wear the same garments in different sizes, Tiwari *et al.* [TBTP20] propose a SizerNet to approximate the wearing effect of a garment in different sizes. Because the dataset only consists of A-pose garments and garments, the proposed method cannot generate a variety of deformations in different poses. To solve the garment-body interpenetration, novel garment space is proposed in Santesteban *et al.* [STOC21], which eliminates the need for any postprocessing steps. Although these

studies have achieved success in the automatic clothing deformation approximation with fully connected layers, a common limitation of these methods is the generalization ability, *i.e.*, independent training is always required when deforming new garments with new mesh topologies.

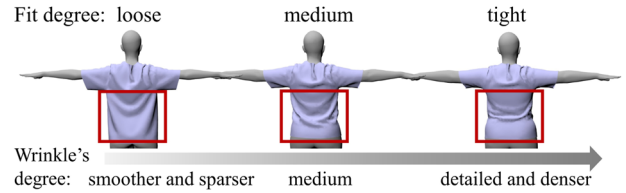
To address the fundamental limitation of generalization in learning-based deformations, research tries to approximate the clothing deformation using graph neural networks which can handle 3D data in non-Euclidian domains. The latest study in Chentanez *et al.* [CMM\*20] introduces a graph neural network with a novel convolution operator for cloth and body skin deformation approximation. The proposed solution is specifically for triangle meshes. Inspired by point cloud processing, Gundogdu *et al.* [GCS\*19] introduce the framework based on PointNet for clothing animation. The results look plausible but tend to be overly smooth. Focusing on fast clothing deformation, Vidaurre *et al.* [VSGC20] present a fully convolutional graph neural network (FCGNN) to predict deformations with fine-scale details. The framework consists of two graph neural networks with the same structure and a different number of layers, which respectively predict the coarse draping and refinement. The proposed pipeline can generalize to unseen mesh topologies, garment parameters and body shapes. However, the prediction is only for one pose and does not consider pose variations. Bertiche *et al.* [BMTE21] predict clothing deformations by using GCN and MLP together, but are unable to achieve satisfactory results for loose garments, such as dresses.

Alternatively, recent research [CPA\*21] introduces an SMPLicit model for garments using an implicit representation that is capable of representing garments with different topologies. However, some detailed deformations cannot be achieved precisely, especially for loose garments. Other studies achieve clothing animation from the perspective of computer vision. Research in Deep-Wrinkles [LCT18] learns a conditional adversarial network to generate high-frequency details in normal maps. Recently, Zhang *et al.* [ZWCM21] tackle the generalization problem and make it possible to transfer details across the normal maps of different garments. Realistic clothing animation can also be achieved by the deforming garment with the displacement map [JZGF20]. However, it falls short when applied with loose garments for the body.

**Learning-based deformation.** Several methods also apply data-driven models to deformation approximation for animated characters. Loper *et al.* [LMR\*15] present a learned skinned multi-person linear model (SMPL) of human body shape and pose-dependent shape variation. Based on this, dynamic blend shapes are predicted in Refs. [CO18, SGOC20] to enrich soft-tissue effects. These approaches can be well generalized to new shapes and motions, but only work for body meshes with a fixed number of vertices. By mapping nodal linear deformations to the nonlinear one with contextual features, Luo *et al.* [LSW\*18] can achieve elastic body simulation in real-time. To make film-quality characters run at interactive rates, Bailey *et al.* [BODO18] train multiple MLPs for one specific character. The generalization problem is solved in Liu *et al.* [LZT\*19], which uses graph neural networks to predict skinning weights for game characters with complicated dressing. Research in Xu *et al.* [XZK\*20] also utilizes the graph neural network to predict the number of joints and skinning weights. To achieve realistic deformation, the non-linear corrections are predicted in each pose step [LSK20, LSK21] by using the improved graph neural networks. Inspired by these methods, in this work, we adopt the SMPL model as the base body and design graph-learning-based models to achieve the clothing deformation with good generalization ability and high-quality results.

### 3. Overview

Given a garment with arbitrary mesh topology, a target human body with any shape, and a series of poses in motion, our goal is to automatically generate realistic clothing deformation with fine-scale wrinkles. Training and predicting this task are not simple due to the high variance of the deformation details. To address this challenge, we first propose a fit attribute that can affect the details of wrinkles to a large extent (Section 4.1). Together with shape and pose attributes, the multi-source attributes enable us to predict more realistic clothing deformations and can give the model good generalization capabilities. Next, to fundamentally mitigate the complexity of the task, while ensuring high-quality deformation effects, we propose a new perspective of output reconstruction (Section 4.2). Unlike the direct prediction of the displacement of each vertex in all previous studies, we decompose this displacement so that the numerical range of the prediction target is greatly reduced. With these strategies, we introduce a pipeline that divides the deformation into two steps. The first step (Section 4.3) is to learn a coarse garment generator to globally produce smooth clothing deformations with global draping effects.



**Figure 3:** Fit of garment and body influences wrinkles.

As depicted in Figure 2, we use a coarse garment generator  $W_{\text{coarse}}$  to achieve this, where  $W_{\text{coarse}}$  is designed with two branches consisting of graph-attention-based (GAT) blocks and fully connected layers. Next, the second step (Section 4.4) is to further enhance details based on the coarse garment. Because of the complexity of this step, as shown in Figure 2, we design an attribute parser  $W_{\text{parser}}$  to generate detail-aware encodings based on multi-source attributes and then infuse them into detail garment generator  $W_{\text{detail}}$  to generate rich and plausible wrinkles locally. With the help of  $W_{\text{parser}}$ , excessive smoothness can be avoided in deformations generated by  $W_{\text{detail}}$  to a certain extent.

## 4. Approach

### 4.1. Garment-body descriptor

To achieve complex clothing deformations, we first observed parameters that affect the quality of deformations. In real scenes, when the relationship between clothes and body (*i.e.*, the degree of fit) varies, the effect of garments on both global (rough) and local (detailed) deformation is also different. As shown in Figure 3, for the fixed material, when the fit degree is from loose to tight, the wrinkles of garments are from smoother and sparser (with a wider wrinkle width) to finer and denser (with a narrower wrinkle width). This observation demonstrates the need to generate the fit parameter as one of the network inputs, helping to better target the different fits of garments to produce more realistic deformations. Next, we will describe how to build this relationship between garment and body and how to express this variation.

For the target body, we adopt the SMPL [LMR\*15] model which represents the human body  $M_b$  with  $N_b$  vertices parameterized by shape ( $\beta$ ) and pose ( $\theta$ ):

$$M_b = W_{\text{smpl}}(\bar{M}_b(\beta, \theta), J(\beta), \theta, \mathcal{W}), \quad (1)$$

$$\bar{M}_b = T + B_s(\beta) + B_p(\theta), \quad (2)$$

where the learned skinning function  $W_{\text{smpl}}(\cdot)$  is applied to deform the rest-pose mesh  $\bar{M}_b(\beta, \theta)$  with skinning weights  $\mathcal{W}$  of the skeleton  $J(\beta)$ .  $\bar{M}_b(\beta, \theta)$  is computed by applying shape blend shapes  $B_s(\beta)$  and pose blend shapes  $B_p(\theta)$  to the mean template mesh  $T \in \mathbb{R}^{N_b \times 3}$ .

Given the SMPL body  $M_b \in \mathbb{R}^{N_b \times 3}$  and garment  $M_g \in \mathbb{R}^{N_g \times 3}$ , we next explore their correspondence. We define the indicator matrix  $\mathbf{I} \in \{0, 1\}^{N_g \times N_b}$  to indicate whether a garment vertex is associated with a body vertex, where the indicator matrix  $\mathbf{I}$  is obtained by finding the closest vertex from garment to body. Here, we assume that the body mesh has sufficient resolution and allows for the

one-to-one correspondence between body vertices and garment vertices. For each garment–body pair  $(\bar{M}_g, \bar{M}_b)$  in rest pose, the distance vector between corresponding garment vertices to body vertices can be calculated as  $\mathbf{d} = \|\bar{M}_g - \bar{M}_b\|$ , where  $\mathbf{d} \in \mathbb{R}^{N_g}$ , and  $\|\cdot\|$  denotes the Euclidean norm operation along the last dimension of the vertex matrix.

Next, we need to concatenate the distance vectors of all garment–body pairs into a matrix, in preparation for exploring a concise representation of the distance information in each pair. However, the ‘vector concatenation’ here is difficult because the vector length  $N_g$  varies in different garment–body pairs. Therefore, to make all the distance vectors of the same length so that they can be concatenated together, based on the minimum number of vertices  $N_g^*$  in the dataset, a fixed number of  $N_g^*$  elements are selected evenly from the distance vector  $\mathbf{d}$  of each garment–body pair to form the new distance vector  $\mathbf{d}^*$ , where  $\mathbf{d}^* \in \mathbb{R}^{N_g^*}$ . Specifically, the selection is accomplished by rejecting  $(N_g - N_g^*)$  vertices closer than the radius  $r = \sqrt{A_g/(CN_g^*)}$  where  $A_g$  is the area of a garment mesh. If not enough vertices are returned, the radius is gradually reduced by increasing the integer  $C$  until the number of vertices equals  $N_g^*$ . With the fixed length distance vectors, we can then concatenate them to form a distance matrix  $\mathbf{D} = [\mathbf{d}_1^*, \dots, \mathbf{d}_{N_{\text{pair}}}^*] \in \mathbb{R}^{N_g^* \times N_{\text{pair}}}$  which stores distance information between all garment–body pairs. Notice that the strategy of fixing  $N_g^*$  is only used here when constructing the distance matrix, while in the latter sections, the networks are still input to garments with an arbitrary number of vertices  $N_g$ .

Next, we seek a parametric expression to represent this information concisely. We compute the fit parameter using factor analysis (FA) to model the variance along each vertex independently. Considering the speed of convergence, we use SVD-based likelihood optimization [SLY20]. FA in matrix term is defined as:

$$\mathbf{D} - \mu \approx \mathbf{L}\mathbf{A}, \quad (3)$$

where  $\mu \in \mathbb{R}^{N_g^*}$  is the mean vector which should be broadcast to the same size as  $\mathbf{D} \in \mathbb{R}^{N_g^* \times N_{\text{pair}}}$  for the subtraction.  $\mathbf{L} \in \mathbb{R}^{N_g^* \times F}$  and  $\mathbf{A} \in \mathbb{R}^{F \times N_{\text{pair}}}$  denote the loading matrix and factors. In this way,  $\mathbf{A}$  consists of  $N_{\text{pair}}$  of vector  $\alpha = [a_1, a_2, \dots, a_F] \in \mathbb{R}^F$  which provides an efficient  $F$ -dimensional representation for each garment–body pair. We call this parameter  $\alpha$  as the fit attribute. At runtime, given the test garment–body pair in rest pose, we use the trained FA model to perform matrix multiplication only once (in rest pose) to directly obtain the fit relationship  $\alpha$ .

In addition, for clothing deformation, body shape and pose also have an impact on the detail folds. Hence, we refer to these three parameters  $(\alpha, \beta, \theta)$  collectively as multi-source attributes. These multi-source attributes play a key role in generating detailed deformations, which are taken as the input of  $W_{\text{parser}}$  introduced in Section 4.4.

## 4.2. Output reconstruction

Most deformation approximation studies are plagued by the problem of highly nonlinear output, *i.e.*, vertex position or displacement. For the output of each vertex, the value of each element in the output vector ranges from negative infinity to positive infinity, leading some studies to utilize only a large number of

fully connected layers while sacrificing generalization [PLPM20], or to make predictions for only one pose for quality assurance [VSGC20]. So far, there has been no research attempting to solve the problem fundamentally from the perspective of reconstructing output.

In our work, we propose an output reconstruction method by decomposing the output vector of each vertex into the magnitude and direction:

$$\Delta_i = \|\Delta_i\| \odot \delta_i, \quad (4)$$

where the original output is  $\Delta_i \in \mathbb{R}^3$ , the decomposed magnitude is  $\|\Delta_i\| \in \mathbb{R}^+$  and the direction is  $\delta_i \in \mathbb{R}^3$ . The operator  $\odot$  means Hadamard product, where the magnitude  $\|\Delta_i\|$  should be broadcast to the same size as the direction  $\delta_i$  for element-wise multiplication. Unlike other learning-based methods which directly predict  $\Delta_i$  with a wide value range of  $(-\infty, +\infty)$  of each dimension of the vector, our method indirectly predicts the vector’s magnitude  $\|\Delta_i\|$  with the narrow value range of  $[0, +\infty)$ , and the direction vector  $\delta_i$  with each dimension value range of  $[-1, 1]$ . In our two generators (shown in Figure 2), both networks are designed with two branches in order to predict the decomposed items separately. In addition, based on the value characteristics, we adopt different activation functions in two branches: ReLU is used in the  $\|\Delta_i\|$  branch to output positive values; Tanh is used in the  $\delta_i$  branch to map the resulting values between  $-1$  and  $1$ . Thus, in contrast to the original output  $\Delta_i$  with the infinite degree of freedom, the value range of our decomposed output is greatly ‘narrowed’, and with the help of the activation function, it can be ensured that the output is always within a reasonable range.

With the two approximated items of  $\|\Delta_i\|$  and  $\delta_i$ , we finally multiply them together to obtain the final non-linear offset vector. The decomposition step does not seem complicated, and it plays a crucial role that greatly mitigates the complexity of learning and can generate better quantitative and qualitative results.

## 4.3. Coarse garment prediction

As stated in previous work [SOC19, PLPM20, VSGC20], directly regressing clothing deformations as a function of designed parameters with one model will result in unrealistic results. Therefore, the final deformation process must be divided into several steps to perform approximations. In this work, we also decompose clothing deformation into coarse deformations with the overall fit and detailed deformations with fine-scale wrinkles.

The goal in the first step is to achieve plausible clothing coarse deformation  $M_{\text{coarse}}$  in the garment worn by the target body  $M_b$  in a certain animated pose:

$$M_{\text{coarse}} = \mathbf{I}M_b + \Delta_{\text{coarse}}, \quad (5)$$

where  $\mathbf{I} \in \{0, 1\}^{N_g \times N_b}$  refers to the indicator matrix of the association between garment and body vertices. For the remaining residual part  $\Delta_{\text{coarse}} \in \mathbb{R}^{N_g \times 3}$ , we aim to learn a model  $W_{\text{coarse}}$  to automatically infer the offsets.

With the garment and animated body, we first need to construct a parametric space that can concisely express useful information for coarse deformation without ignoring the spatial information.

Therefore, we consider the input of our network to be a graph. Through the indicator matrix  $\mathbf{I}$ ,  $N_{b'}$  body vertices are associated with  $N_g$  garment vertices, where  $N_{b'} = N_g$ . Based on these body vertices, we then construct the graph:  $\mathcal{G}_{b'} = (\mathcal{V}^{b'}, \mathcal{E}^{b'})$  which stores vertex features  $\mathcal{V}^{b'}$  of  $N_{b'}$  body vertices and their edges  $\mathcal{E}^{b'}$  where  $(i, j) \in \mathcal{E}^{b'}$  denotes an edge connection between a node  $i$  and a node  $j$ . In particular, the connection represented by  $\mathcal{E}^{b'}$  is equal to the connection of the garment vertices. Next, for each node, we need to assign attributes to make the node informative. Specifically, to encode the body mesh geometry, we append the vertex normal  $n_i^{b'} \in \mathbb{R}^3$  to each graph node; to reflect the body skinning features in different poses, we adopt the relative skinning features [LSK21]  $p_i^{b'} = \sum_{s=1}^S w_{s,i} G_s(\theta) G_s(\theta^*)^{-1} \bar{p}_i^{b'} \in \mathbb{R}^3$  to each node features, where  $w_{s,i}$  is the skinning weight of the vertex  $i$  affected by the joint  $s$ ,  $G_s(\theta)$  is the rotation matrix of joint  $s$  in pose  $\theta$  ( $\theta^*$  denotes the rest pose) and  $\bar{p}_i^{b'}$  is the rest pose position. Specifically, this relative skinning feature is a variant of the body vertex position, *i.e.*, when the body is moved to an arbitrary location (no rotation), the body vertex position changes while the relative skinning feature remains the same. Since body features  $n_i^{b'}$  and  $p_i^{b'}$  alone cannot predict clothing behaviours, we need to attach fit attributes to each node to represent the relationship between the body and the garment. Here, for simplicity and conciseness, the first component  $a_1$  of the fit attribute  $\alpha$ , which is the most discriminative one, is adopted. In total, each node feature  $v_i^{b'}$  in  $\mathcal{V}^{b'}$  consists of three attributes, which can be expressed as:  $v_i^{b'} = [n_i^{b'}, p_i^{b'}, a_1] \in \mathbb{R}^7$ .

Having the graph with defined features as input, next we need to design a model  $W_{\text{coarse}}$  for acquiring the latent representation of the graph data and mapping it to the final prediction  $\Delta_{\text{coarse}}$ . To accomplish this task, there are two requirements for the design model  $W_{\text{coarse}}$ . Specifically, first, the model should have the generalization ability that is able to deal with garments with arbitrary mesh topologies. Second, the model should be able to infer the overall deformation of garments under various body shapes and postures according to the knowledge learned in the training process. To satisfy these needs at the same time, we adopt GAT blocks which extend the original GAT structure [VCC\*17] with the self-reinforced stream [LSK20] for efficiently handling complicated 3D mesh features. Specifically, the original GAT structure computes hidden representation of the graph node by aggregating the weighted neighbouring features; moreover, the self-reinforced stream uses a fully connected layer to linearly map the original node features to the latter layer. By aggregating node features from the neighbourhoods and strengthening self-features, such GAT blocks allow for acquiring the latent representations of irregular mesh graph data without the need of knowing the graph structure upfront.

As shown in Figure 2, for coarse deformation prediction, first we apply one block in the first layer for dealing with the input graph, and then apply three blocks to each branch, *i.e.*, the magnitude prediction branch and the direction prediction branch. The reason for designing two branches is that the value range of two predictions (as stated in Section 4.2) is different and each branch needs to adopt a different activation function to ensure the range of the output value. In the last layer of two branches, linear transformation and corresponding activation and normalization are used therefore achieving the final predictions: the magnitude  $\|\Delta_{\text{coarse}}\|$  and the direction

$\delta_{\text{coarse}}$ . The whole progress through the coarse generator can be expressed as:

$$\|\Delta_{\text{coarse}}\|, \delta_{\text{coarse}} = W_{\text{coarse}}(\mathcal{G}_{b'}). \quad (6)$$

Lastly, two predictions are element-wisely multiplied together to get the displacement  $\Delta_{\text{coarse}}$  to the body. During the training, we minimize the MSE loss between the predicted displacement  $\Delta_{\text{coarse}}$  and the ground truth  $\Delta_{\text{coarse}}^{\text{GT}}$ .

#### 4.4. Detail garment prediction

After obtaining the coarse clothing deformation, the next step is to realize detailed deformation with fine-scale wrinkles. Compared with coarse deformation that is easy to generate, detailed deformation is extremely difficult to obtain due to its complexity and volatility under various states. Despite research advances, existing learning-based studies always have to face the trade-off between the generalization ability of models and the fidelity of results, *i.e.*, the model is only worked for the specific mesh topology [PLPM20] or for rest pose [VSGC20, TBTP20] and tends to produce overly smooth deformations [GCS\*19]. Even though significant efforts have been made on many aspects such as input improvement, network structure improvement, convolution operator change and increase in the number of models, different degrees of wrinkles in diverse poses and shapes still cannot be stably learned and approximated.

To address these challenges, we propose the novel detail-aware attribute parser  $W_{\text{parser}}$  and detail garment generator  $W_{\text{detail}}$ , where the key idea is to adjust the wrinkle-related adaptive distribution of the graph and transfer it through two branches for detailed deformation approximation.

On one hand, given the generated coarse deformation, we build a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , in which  $\mathcal{V} = \{v_1, \dots, v_{N_g}\}$  indicates clothing mesh node features, and  $\mathcal{E}$  is mesh edges. For each node, the features are defined as:  $v_i = [n_i, p_i, x_i]$ , which consists of the vertex normal  $n_i \in \mathbb{R}^3$ , the relative skinning features  $p_i \in \mathbb{R}^3$  (as stated in Section 4.3) and the distance vector from clothing vertices to all joints  $x_i = [x_{i,1}, \dots, x_{i,S}] \in \mathbb{R}^S$  ( $S$  is the number of joints).

On the other hand, given a series of attributes that affect the degree of wrinkles, directly constructing graphs by assigning attributes (*e.g.*, shape, pose, *etc.*) to every single node and then forwarding them into the network is the most common strategy in previous graph-learning-based methods. However, it will lead to feature redundancy because attributes are independent on a single node. Therefore, we design a detail-aware attribute parser (as shown in Figure 2) that takes the multi-source attributes  $(\alpha, \beta, \theta)$  as input and the detail-aware encodings  $W_{\text{parser}}(\alpha, \beta, \theta)$  as output that can adaptively adjust the graph feature distribution based on a given input instance. Specifically, detail-aware encodings are vectors where their dimensions equal to  $d^{[1]}$ , *i.e.*, the dimension of the graph feature of each vertex after the first layer. Then, we element-wisely multiply it with the transformed graph along the feature dimension:

$$\mathcal{G}^* = W_{\text{parser}}(\alpha, \beta, \theta) \odot W_{\text{detail}}^{[1]}(\mathcal{G}), \quad (7)$$

where  $\mathcal{G}^*$  refers to the graph with the infused features after the first layer of the graph  $W_{\text{detail}}^{[1]}(\mathcal{G})$  and the detail-aware encodings



**Figure 4:** Training and test samples in our dataset. Note that all test samples are unseen in the training set.

$W_{\text{parser}}(\alpha, \beta, \theta)$ . In other words, the original features in  $W_{\text{detail}}^{[1]}(\mathcal{G})$  have been adaptively modified by high-dimensional attribute encodings, so that new features in  $\mathcal{G}^*$  can be expressed in a more detail-aware manner and be prepared for accurate prediction.

We input the new graph  $\mathcal{G}^*$  into the following layers of  $W_{\text{detail}}$  (except for the first layer  $W_{\text{detail}}^{[1]}$ , the remaining part can be expressed as  $W_{\text{detail}}^{[2-L]}$ ). Similar to the coarse generator  $W_{\text{coarse}}$ , the detail generator  $W_{\text{detail}}$  also has two branches, which respectively approximate decomposed detail output elements: the magnitude  $\|\Delta_{\text{detail}}\|$  and the direction  $\delta_{\text{detail}}$ . Due to the difficulty of the detailed deformation approximation, for each branch, in addition to graph-attention-based blocks, we also apply graph pooling and unpooling operations [Die19] to avoid over-fitting problem and improve the model generalization ability. In conclusion, the approximation via the detail generator after the first layer  $W_{\text{detail}}^{[2-L]}$  can be expressed as:

$$\|\Delta_{\text{detail}}\|, \delta_{\text{detail}} = W_{\text{detail}}^{[2-L]}(\mathcal{G}^*). \quad (8)$$

We multiply the predicted  $\|\Delta_{\text{detail}}\|$  and  $\delta_{\text{detail}}$  to obtain the corrective displacement  $\Delta_{\text{detail}}$ , and add this to the coarse deformation to obtain the ultimate detailed clothing deformation:

$$M_{\text{detail}} = M_{\text{coarse}} + \Delta_{\text{detail}}. \quad (9)$$

During the training process,  $W_{\text{parser}}$  and  $W_{\text{detail}}$  are optimized simultaneously. We adopt MSE loss as the loss function to minimize the difference between the predicted  $\Delta_{\text{detail}}$  and the ground truth  $\Delta_{\text{detail}}^{\text{GT}}$ .

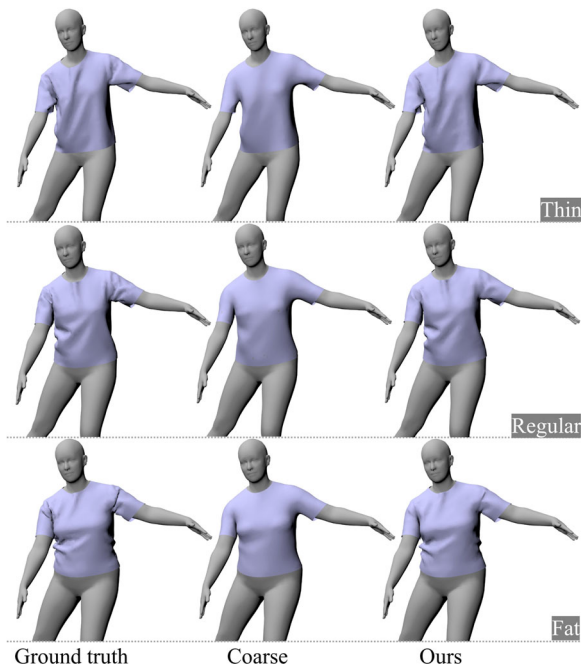
## 5. Evaluation

### 5.1. Dataset and Implementation

To evaluate our proposed method, we create a dataset (Figure 4) consisting of various garments, body shapes and animated poses for training and testing. To produce the ground truth data of garments, we utilize the 3D clothing design and simulation software Marvelous Designer [Mar] to design and generate clothing deformations different mesh topologies and the number of vertices. To obtain the coarse data, as in Refs. [VSGC20, PLPM20], we also apply the Laplacian smoothing operator (with 0.12 diffusion coefficient and

40 iterations) to each generated clothing mesh. Then, to generate different bodies, we adopt SMPL parametric human model and sample the second and seventh shape components. The original SMPL template has 6890 vertices, which we re-mesh to give it sufficient mesh resolution (with 27,554 vertices) to achieve one-to-one correspondence with the garment mesh. This is done by applying 4-to-1 sub-division once for each triangle of an original mesh. For the pose variation, we select animated poses from CMU mocap [CMU] and AMASS dataset [MGT\*19], including motion sequences of dancing, ballet, *etc.* In particular, we divide the dataset into a training set, a test set and a validation set, and ensure that the data in them do not overlap. In our training set, we use 17 garments and six bodies with 2907 poses for each garment-body pair. Then, to verify the effectiveness of the methods, we use seven garments, three body shapes and 405 poses in the test set. Further, to effectively help us keep track of training progress, we adopt three garments, two bodies and 103 poses for validation.

For the implementation, as shown in Figure 2, we next describe the detailed structure of  $W_{\text{coarse}}$ ,  $W_{\text{parser}}$  and  $W_{\text{detail}}$ . For training  $W_{\text{coarse}}$ , the features of graph  $\mathcal{G}_{b'}$  are input into a GAT block with the hidden feature size of 256 where the multi-head number is 4, the feature sizes of the self-reinforced stream and aggregation stream are 128 and 32, respectively. Features are applied with Tanh activation and then fed into the  $\|\Delta_{\text{coarse}}\|$  prediction branch and the  $\delta_{\text{coarse}}$  prediction branch, both branches contain three GAT blocks with the hidden feature size of [512, 512, 256]. After graph convolution, three fully connected layers are used to transform the features with the hidden sizes of [256, 128, 1] in the  $\|\Delta_{\text{coarse}}\|$  prediction branch and of [256, 128, 3] in the  $\delta_{\text{coarse}}$  prediction branch. To ensure that the output range is reasonable, ReLU and Tanh activation functions are used, respectively, after each layer of the two branches. Additionally, normalization is also used for features in the  $\delta_{\text{coarse}}$  branch. For training  $W_{\text{parser}}$ , the multi-source attributes  $(\alpha, \beta, \theta)$  (where  $\alpha \in \mathbb{R}^3$ ,  $\beta \in \mathbb{R}^{10}$ ,  $\theta \in \mathbb{R}^{72}$ ) are transformed into detail-aware encodings by three fully connected layers ([256, 512, 1024]) and ReLU activation function. For training  $W_{\text{detail}}$ , the graph features  $\mathcal{G}$  are fed a GAT block with the hidden feature size of 1024. After infusing graph features with detail-aware encodings, the feature dimension is unchanged and features are input into the  $\|\Delta_{\text{detail}}\|$  prediction branch



**Figure 5:** Generalization to new thin, regular and fat bodies.

and the  $\delta_{\text{detail}}$  prediction branch. The structure of four GAT blocks ([256, 256, 128, 96]) and operations of pooling ( $N_g$  roughly becomes half) and unpooling (restored) are the same in each branch. Finally, fully connected layers with the hidden feature sizes of [128, 64, 1] and [128, 64, 3] and corresponding activations are, respectively, adopted in the  $\|\Delta_{\text{detail}}\|$  prediction branch and  $\delta_{\text{detail}}$  prediction branch.

## 5.2. Quantitative and qualitative evaluation

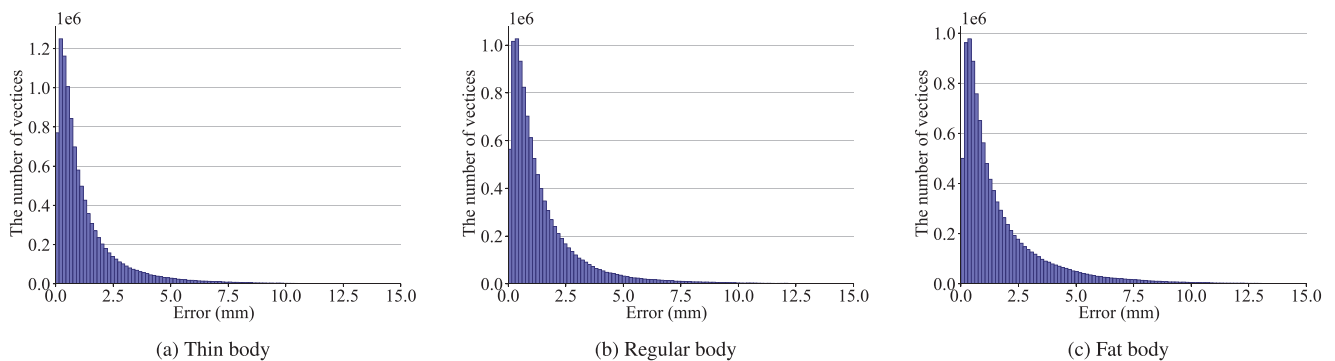
**Generalization to new bodies.** As shown in Figure 5, we provide the generalization results of thin, regular and fat bodies that are unseen in the training set. Based on the predicted coarse deformation, our method is able to generate fine-scale wrinkles which have no ob-

**Table 1:** Mean error (mm) of per-vertex deformations in different body shapes.

Test shapes	Thin	Regular	Fat
Coarse	2.82	3.01	3.27
Detail	1.33	1.52	1.74

vious difference with the ground truth data. In addition, our method can successfully predict individualized and detailed clothing deformations of bodies with different shapes, which contains rich and plausible wrinkles in the area of the left side of the waist. During the training, the influencing attributes are transferred into detail-aware encodings and clothing deformation is learned in a detail-aware manner, so that we can effectively make accurate predictions for new body shapes. Quantitatively, in Figure 6, we counted the error distribution of these three test bodies wearing the same training garments under the same training poses. As observed, the number of vertices is the highest in the clothing deformation errors of thin bodies close to zero (Figure 6a), and the mean error of per vertex is about 1.33 mm as reported in Table 1. The deformation prediction error of the garment worn by thin bodies is relatively smaller since the clothing folds are simpler than the garment worn by fat bodies; in contrast, the garment worn by fat bodies has more complicated folds, making them relatively difficult to predict. Overall, through the deformation refinement of  $W_{\text{detail}}$  and  $W_{\text{parser}}$ , deformation errors are reduced by about half compared with coarse deformations.

**Generalization to new poses.** Figure 7 shows the results of visually evaluating the quality of our proposed approach of generalization to new poses, in which we compared the deformations of the ground truth physics-based simulations and our predictions. We animated dressed bodies with new postures of raising the hand, walking and swinging. Through the proposed method, attractive details can be successfully generated, in which the wrinkles in areas of armpits, waist, shoulders are rich and quite similar to real effect of the ground truth with per-vertex prediction error of 1.67 mm. During the training, in addition to graph constructions, we also design a  $W_{\text{parser}}$  to generate detail-aware encodings and infuse them into the graph



**Figure 6:** Histogram plot of distribution of per-vertex errors of generalization to new body shapes. The bin width is 0.15. The first two bars (error: 0–0.15 and 0.15–0.3) in (a) have the largest number of vertices, while those in (c) have the smallest number of vertices.



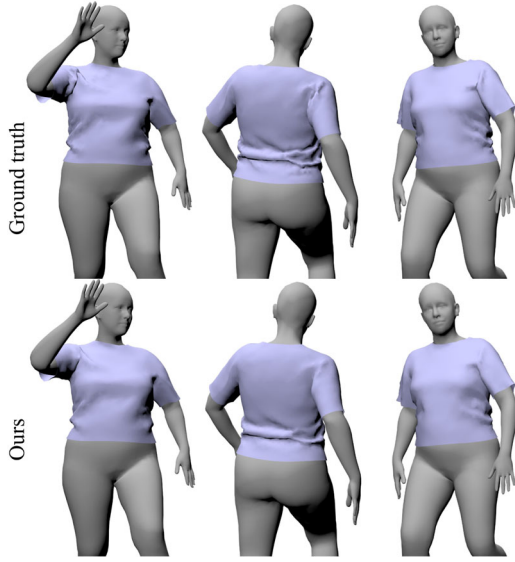


Figure 7: Generalization to new poses.

neural network, so that the model can learn the individualized deformations caused by different poses.

**Generalization to new garments.** Figure 8 shows the qualitative results of the generalization to new garments, *i.e.*, long t-shirt and vest. Here, the test garments have different garment meshes and number of vertices from the training. Thanks to the graph-learning-based model and proposed detail-aware strategies, our model can reasonably approximate deformations with rich details regardless of the garment design. Due to the influence of the hem, long t-shirt fits tightly to the body compared to the vest, so more dense wrinkles appear in the deformation results around areas of stomach and waist, which also follows the law of our first observation as stated in Section 4.1.

**Generalization to new garments, bodies and poses simultaneously.** Figure 9 and Table 2 show our results on unseen garments,

Table 2: Mean error (mm) of per-vertex deformations in unseen garments, bodies and poses.

Test	Coarse	Detail
Dress + thin body + new poses	3.46	2.11
Jacket + regular body + new poses	3.65	2.39
Coat + fat body + new poses	3.63	2.27

bodies and poses at the same time. Specifically, the types of test garments have different mesh topologies, including cut-out detail dress, short sleeve jacket and 3/4 sleeve coat. At the same time, we let characters with new body shapes wear these garments and perform animations with new poses. Despite the fact that all three variables are brand new and do not appear in the training set, our predictions still naturally match the ground truth and most of the fine-scale wrinkles can be successfully produced. Overall, the per-vertex average error of the predictions for all the test data is about 2.24 mm. Results demonstrate that our proposed method has powerful generalization capabilities to handle completely new variation terms simultaneously, and thus can be easily integrated into practical applications.

### 5.3. Ablation study

We conducted an ablation study to highlight the effectiveness of our strategies: output decomposition, detail-aware attribute parser, two-step approximation and graph pooling operation.

To evaluate the proposed output decomposition, we first retain output displacement of per-vertex as the original three-dimensional vector  $\Delta_{\text{detail}}$  (w/o decomposition), allowing the network to have an unbounded prediction range. Further, we set a limited value range for the three-dimensional output displacement (w/output limit) where the limited range is obtained by scaling each original displacement value to  $(-1, 1)$  using a scale factor of 5 determined by the dataset. Next, to evaluate the detail-aware attribute parser, we tested the case of removing the attribute parser (w/o attribute parser) where attributes are directly assigned to each graph node, and the

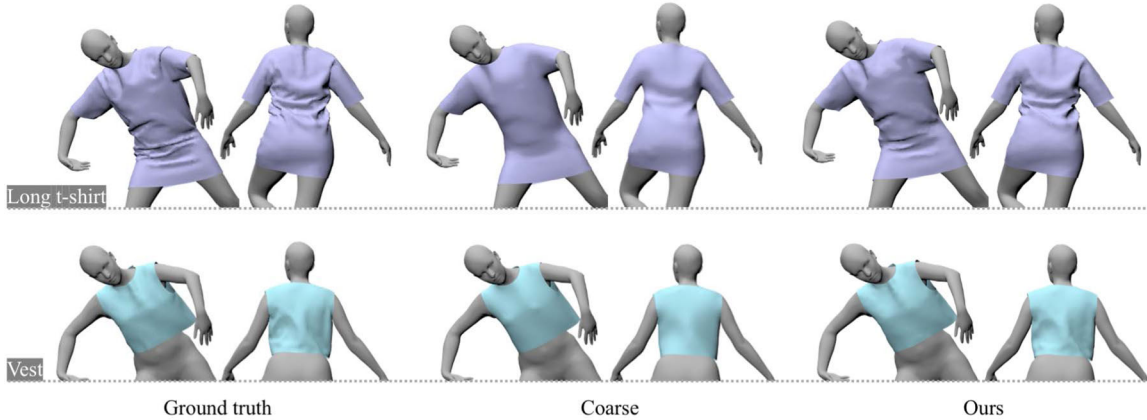
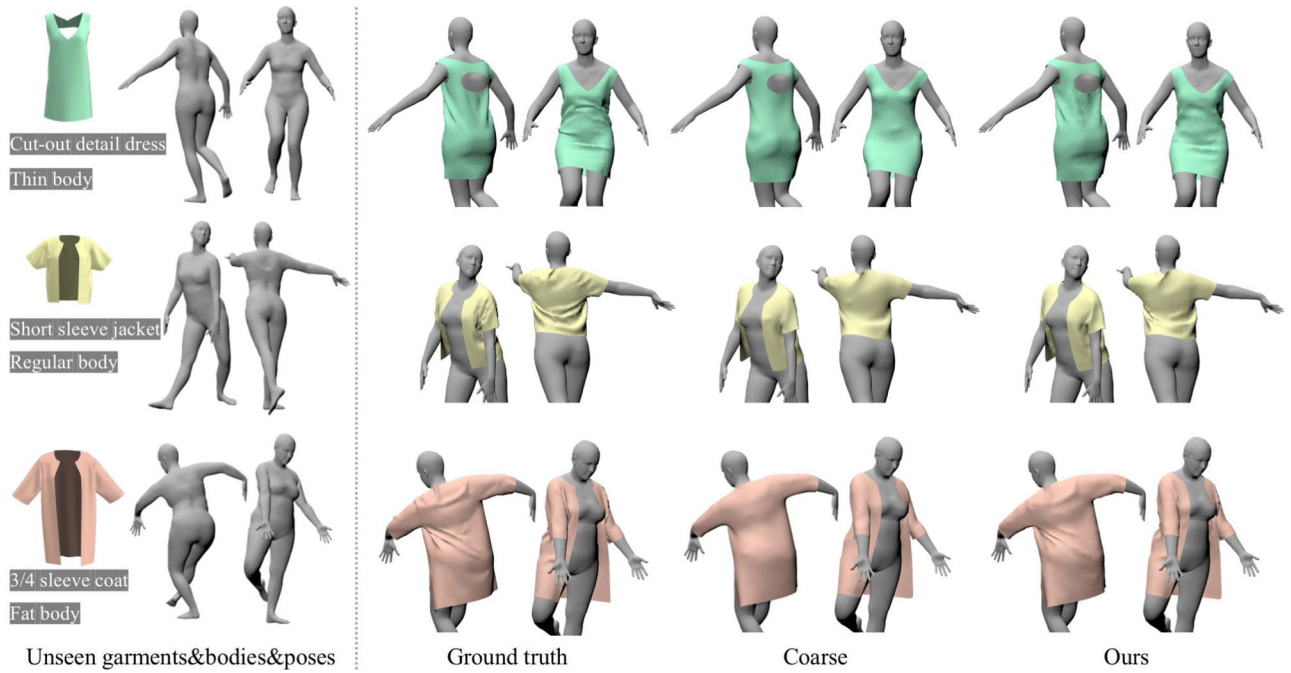


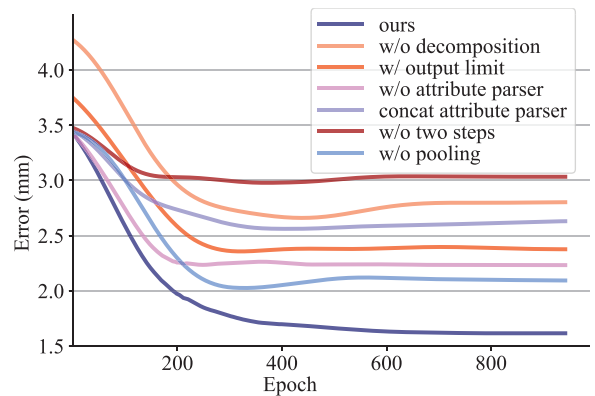
Figure 8: Generalization to new short and long garments.



**Figure 9:** Generalization to unseen garments, bodies and poses simultaneously.

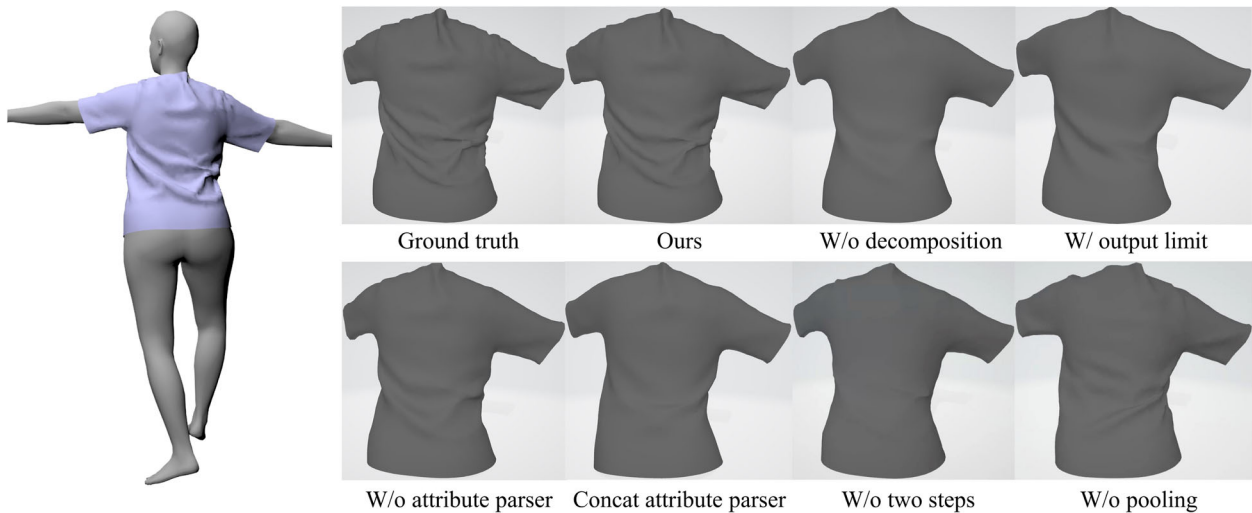
case of replacing the element-wise multiplication with concatenation in Equation (7) (concat attribute parser). Finally, to evaluate the usefulness of two-steps approximation and graph pooling operation, we adopt a single model instead of two-steps approximation (w/o two steps) and removed pooling operators (w/o pooling) separately. Notice that, for as fair as possible evaluations, the models used in the above experiments have comparable capacities. Specifically, in the cases of the removal of layers, the number of parameters in the remaining layers is increased to ensure the approximate consistency of capacity. Additionally, we choose the best-tested initialization scheme for all evaluations, *i.e.*, Glorot initialization for the graph convolutional layers and Kaiming initialization for the other layers.

As shown in Figure 10, we plot the mean error of per-vertex during the validation process. In the beginning, the method without output decomposition produces the largest error because the output result is difficult to be approximated with three values from negative infinity to positive infinity. As the epoch increases, it is still accompanied by highly complicated outputs and errors remaining between 2.5 and 3 mm that cannot be reduced. Next, when setting the output within a limited value range, the error of the prediction becomes lower than in the case of unbounded output, but the convergence is not ideal. We also observe the importance of the proposed attribute parser. Despite using a network structure with approximately the same capacity as the original after removing the attribute parser, the deformation error is still large. Applying concatenation rather than element-wise multiplication in Equation (7) leads to even worse results, as the attribute information cannot be accurately infused into graph features. Also, without the two-step strategy and the pooling operation, quantitative results are affected to varying degrees. Figure 11 shows the qualitative results of these experiments. As it can be seen, the direct prediction without output



**Figure 10:** Mean vertex error during validation of generating detail deformations. Our proposed output decomposition and attribute parser play a key role in the learning of detailed deformation.

decomposition leads to smoothing artifacts. Although the method of scaling down the output to a limited value range can improve the deformation effect, many detailed wrinkles are still lost. For the result of without attribute parser, the deformation has the obvious folds with the wider width in the waist and collar areas, which can reflect a certain degree of wrinkle trend. However, when using concatenation to combine the attribute parser-processed features with graph features, qualitatively the results are much worse. We also find the worse performance for the method without the two-step approximation, it suggests that mixing global and local deformations for learning substantially increases the difficulty of the task. Finally, for the case of removing the pooling operation, the



**Figure 11:** Qualitative results of ablation study comparing detailed deformations of ground truth of physics-based simulation, approximated by our full method, our method without output decomposition, without detail-aware attribute parser, without two steps and without graph pooling.

**Table 3:** Comparison of our approach with other state-of-art learning-based clothing deformation methods. Our method can achieve more functions with smaller model size.

Methods	Verts. variation	Pose variation	Fit variation	Model size
FCGNN	✓	✗	✓	71.0 MB
TailorNet	✗	✓	✓	2.0 GB
Ours	✓	✓	✓	37.4 MB

deformation contains some details, but the position and trend of some wrinkles differ from the ground truth, suggesting that graph pooling works for the generalization of the model, *i.e.*, using the learned information to make valid inferences. In contrast, our full method is able to successfully generate these major and subtle wrinkles and recover detail effects similar to the ground truth thanks to the proposed strategies.

#### 5.4. Comparison

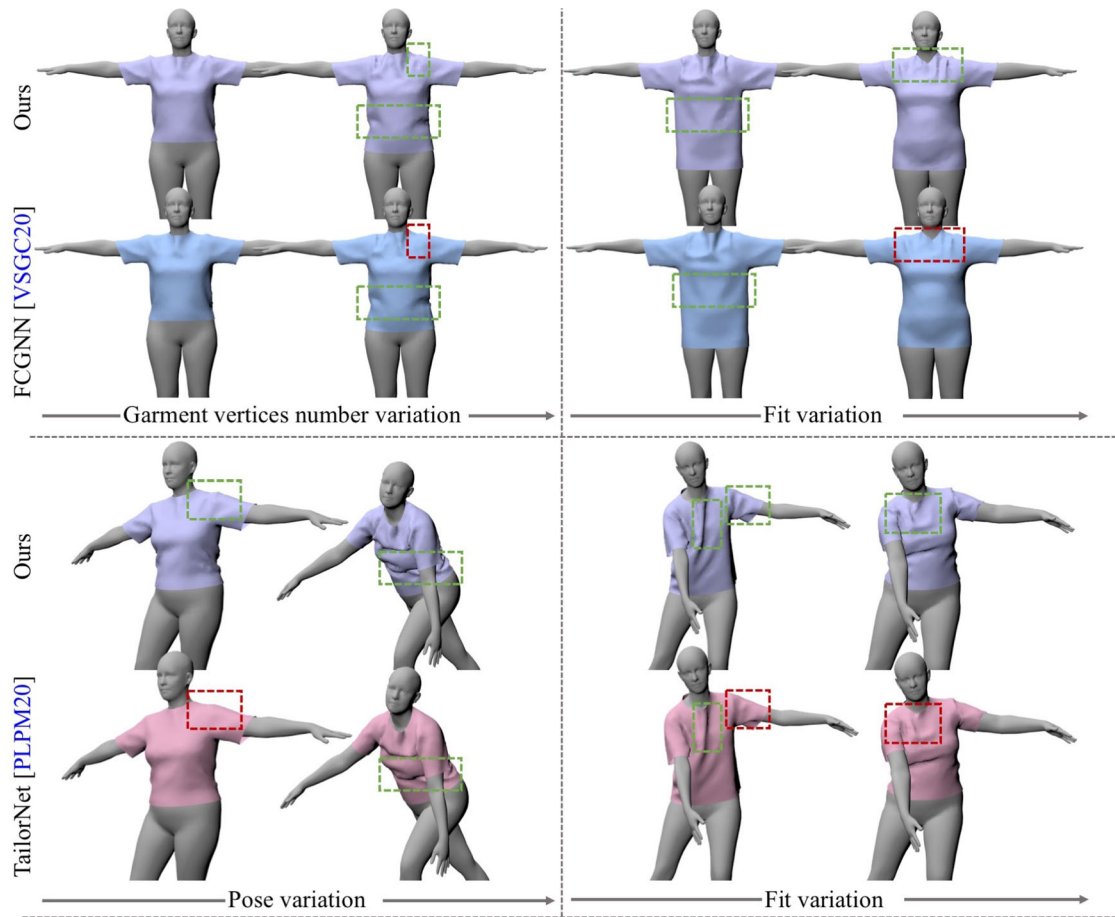
We compare our method with other state-of-art learning-based approaches: FCGNN [VSGC20] and TailorNet [PLPM20]. As listed in Table 3, FCGNN can generalize to arbitrary garment meshes due to the use of a FCGNN, but it only predicts clothing deformations under the t-pose. TailorNet is able to achieve pose-dependent deformations, but it is limited to the use of trained MLP models to predict deformations of new mesh topologies. To the best of our knowledge, currently, there is no prior research involving tasks that are exactly the same as our method to approximate clothing deformations for various mesh topologies and body shapes in diverse poses.

The results of the qualitative comparison are shown in Figure 12 and the results of the quantitative comparison are shown in

**Table 4:** Quantitative comparison of FCGNN [VSGC20], TailorNet [PLPM20] and our method.

Methods	Verts. variation	Pose variation	Fit variation
FCGNN	1.53	-	1.70
TailorNet	-	2.23	2.17
Ours	1.46	1.85	1.56

Table 4. Because of the limited terms listed in Table 3, garments with different number of vertices and garments worn by different body shapes are evaluated for the method of FCGNN [VSGC20]; one garment under different postures and worn by different body shapes are evaluated for the method of TailorNet [PLPM20]. As observed, both FCGNN and TailorNet have the generalization abilities and are capable of generating plausible deformation effects, especially in waist areas, and for garments worn by thin body shapes (as green-framed parts in the figure). Despite the predictions for conspicuous wrinkles, the shoulder areas with small fine-scale wrinkles are still overly smooth (as red-framed parts in the figure). Quantitatively, our method also outperforms previous work. Although both our method and FCGNN choose graph neural networks and both take a two-step approach to deformation prediction, the key to our success in predicting the details of folds under multiple variables lies in the three core techniques we proposed: the fit parameter, the output decomposition and the attribute parser. Without these three techniques, a vanilla graph neural network applied to deformation would suffer from the complexity of multiple variables, and thus, as in the case of FCGNN, would only be able to predict deformation under t-pose and sometimes lose details. For TailorNet, it can successfully generate deformations with some details by over-fitting MLPs for each clothing type with fixed number of vertices. Nevertheless, MLPs require a large number of parameters resulting in the model size of



**Figure 12:** Qualitative comparison of FCGNN [VSGC20], TailorNet [PLPM20] and our method. FCGNN (with blue garments) can achieve mesh topology variation and fit variation; TailorNet can achieve pose variation and fit variation; ours is the first approach to achieve all of these. In addition, our method can generate detail-aware clothing animation, that allows for rich detail prediction caused by various attributes.

around 2.0 GB (just for the t-shirt), which makes them difficult to be applied in practice. Meanwhile, the use of MLPs ignores mesh topology information, resulting in models that do not really learn the detailed deformations and thus the results are sometimes smooth. The comparison shows the benefit of our proposed method: even in the face of multiple variations, the model still has excellent generalization ability to approximate not only obvious wrinkle folds but also fine-scale details.

### 5.5. Runtime performance and memory

With the nVIDIA GeForce RTX2080Ti GPU, in the approximation of clothing deformations of meshes with 3000–4000 number of vertices, the average per-frame runtime is about 21 ms, where coarse prediction takes 8 ms and detail prediction takes 13 ms. The proposed method is 50 times faster than physics-based simulation, making it suitable for real-time applications. The memory footprint of our method is about 37.4 MB, where the  $W_{\text{coarse}}$  model is 18.4 MB and the  $W_{\text{detail}} + W_{\text{parser}}$  model is 19 MB.

## 6. Conclusion

We have presented a graph-learning-based deformation method for garments whose mesh topology can be arbitrary and can be worn by any body shape in various poses. To achieve generalization and high-quality predictions at the same time, we first propose the fit parameter as one of the important attributes influencing the wrinkle details. Then, we design an attribute parser to generate detail-aware encodings and infuse them into the graph neural network to help generate individualized details. Last and most importantly, we propose a novel output reconstruction strategy for the excellent convergence of extremely complex regressions. This strategy can not only be adopted in clothing deformations, but also works for predicting positions or displacement adjustments in other areas. Experimental results have shown that our method with the above technical innovations can overcome the limitations and outperforms existing learning-based approaches.

Despite achieving powerful generalization and impressive detailed deformations, our method still has a few drawbacks that can be addressed in future works. First, we currently use a constant

indicator matrix  $\mathbf{I}$  to represent the correspondence between the garment and the body, but keeping  $\mathbf{I}$  constant causes garment-body collisions when the garment deforms significantly, which is addressed by applying postprocessing step [PLPM20]. In the future, it would be valuable to explore a dynamically updatable indicator matrix and adopting a diffused human model [STOC21] to effectively solve the collision problem. Second, we currently apply SMPL bodies as human models, which provide a parametric space of shape and pose that can be easily used as one of input. In the future when faced with different human model types, our currently input will need to be adapted. For example, the dimension of the pose features  $\theta$  and vertex-joint distance features  $x_i$  is related to the number of joints and therefore is only applicable to rigs with the fixed number of joints. In this situation, the design of skeleton-independent pose features or direct input of body vertex positions is possible solution in future research. Third, each garment in our dataset is given the same material and node distance settings. Future work can expand the dataset to include different materials and tessellation, and then explore related attributes as network inputs to automatically generate more realistic deformations.

### Acknowledgement

This work was partially supported by JSPS KAKENHI (Grant Number JP22K12331).

### References

- [BMTE21] BERTICHE H., MADADI M., TYLSON E., ESCALERA S.: DeePSD: Automatic deep skinning and pose space deformation for 3D garment animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (Montreal, QC, Canada, Oct. 2021), IEEE, pp. 5451–5460. <https://doi.org/10.1109/ICCV48922.2021.00542>.
- [BODO18] BAILEY S. W., OTTE D., DILORENZO P., O'BRIEN J. F.: Fast and deep deformation approximations. *ACM Trans. Graph.* 37, 4 (Aug. 2018), 119:1–119:12.
- [CLMMO14] CIRIO G., LOPEZ-MORENO J., MIRAUT D., OTADUY M. A.: Yarn-level simulation of woven cloth. *ACM Trans. Graph.* 33, 6 (Nov. 2014).
- [CMM\*20] CHENTANEZ N., MACKLIN M., MÜLLER M., JESCHKE S., KIM T.-Y.: Cloth and skin deformation with a triangle mesh based convolutional neural network. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Canada, 2020), Eurographics Association. <https://doi.org/10.1111/cgf.14107>.
- [CMU] CMU graphics lab motion capture database. <http://mocap.cs.cmu.edu/>. (Mar 2009) Accessed: 2021-Aug-21.
- [CO18] CASAS D., OTADUY M. A.: Learning nonlinear soft-tissue dynamics for interactive avatars. *ACM Comput. Graph. Interact. Tech.* 1, 1 (Jul. 2018), 10:1–10:15.
- [CPA\*21] CORONA E., PUMAROLA A., ALENYÀ G., PONS-MOLL G., MORENO-NOGUER F.: Smplicit: Topology-aware generative model for clothed people. In *IEEE Conference on Computer Vision and Pattern Recognition* (June 2021), IEEE, pp. 11875–11885. <https://doi.org/10.1109/CVPR46437.2021.01170>.
- [dASTH10] DE AGUIAR E., SIGAL L., TREUILLE A., HODGINS J. K.: Stable spaces for real-time clothing. *ACM Trans. Graph.* 29, 4 (July 2010). <https://doi.org/10.1145/1778765.1778843>.
- [Die19] DIEHL F.: Edge contraction pooling for graph neural networks. arXiv preprint arXiv:1905.10990 (2019). <http://arxiv.org/abs/1905.10990>.
- [GCP\*22] GUNDOGDU E., CONSTANTIN V., PARASHAR S., SEIFODDINI A., DANG M., SALZMANN M., FUA P.: Garnet++: Improving fast and accurate static 3D cloth draping by curvature loss. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 1 (Mar. 2022), 181–195.
- [GCS\*19] 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/CVF International Conference on Computer Vision (ICCV)* (Seoul, Korea (South), Oct. 2019), IEEE, pp. 8738–8747. <https://doi.org/10.1109/ICCV.2019.00883>.
- [GPV\*15] GILLETTE R., PETERS C., VINING N., EDWARDS E., SHEFFER A.: Real-time dynamic wrinkling of coarse animated cloth. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California, Aug. 2015), ACM Press, pp. 17–26. <https://doi.org/10.1145/2786784.2786789>.
- [JZGF20] JIN N., ZHU Y., GENG Z., FEDKIW R.: A pixel-based framework for data-driven clothing. *Comput. Graph. Forum* 39, 8 (Mar. 2020), 135–144.
- [LC04] LARBOULETTE C., CANI M.-P.: Real-time dynamic wrinkles. In *Proceedings of the Computer Graphics International (CGI)* (Crete, Greece, June 2004), IEEE Computer Society, pp. 522–525. <https://doi.org/10.1109/CGI.2004.1309258>.
- [LCT18] LAHNER Z., CREMERS D., TUNG T.: DeepWrinkles: Accurate and realistic clothing modeling. In *Proceedings of the Computer Vision—ECCV 2018—15th European Conference* (Munich, Germany, Sep. 2018), Springer, pp. 698–715. [https://doi.org/10.1007/978-3-030-01225-0\\_41](https://doi.org/10.1007/978-3-030-01225-0_41).
- [LMR\*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graph.* 34, 6 (Oct. 2015). <https://doi.org/10.1145/2816795.2818013>.
- [LSK20] LI T., SHI R., KANAI T.: DenseGATs: A graph-attention-based network for nonlinear character deformation. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (San Francisco, CA, USA, 2020), ACM, pp. 5:1–5:9. <https://doi.org/10.1145/3384382.3384525>.
- [LSK21] LI T., SHI R., KANAI T.: MultiResGNet: Approximating nonlinear deformation via multi-resolution graphs. *Comput. Graph. Forum* 40, 2 (2021), 537–548.

- [LSW\*18] LUO R., SHAO T., WANG H., XU W., CHEN X., ZHOU K., YANG Y.: NNWarp: Neural network-based nonlinear deformation. *IEEE Trans. Vis. Comput. Graph.* 26, 4 (Oct. 2018), 1745–1759.
- [LZT\*19] LIU L., ZHENG Y., TANG D., YUAN Y., FAN C., ZHOU K.: NeuroSkinning: Automatic skin binding for production characters with deep graph networks. *ACM Trans. Graph.* 38, 4 (July 2019), 114:1–114:12.
- [Mar] Marvelous designer. <https://www.marvelousdesigner.com/>. Accessed: 2022-Mar-10.
- [MC10] MÜLLER M., CHENTANEZ N.: Wrinkle meshes. In *Proceedings of the 2010 Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (Madrid, Spain, May 2010), Eurographics Association, pp. 85–91. <https://doi.org/10.2312/SCA/SCA10/085-091>.
- [MGT\*19] 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/CVF International Conference on Computer Vision* (Seoul, Korea (South), Oct. 2019), IEEE, pp. 5442–5451. <https://doi.org/10.1109/ICCV.2019.00554>.
- [NKT15] NI X., KALE L. V., TAMSTORF R.: Scalable asynchronous contact mechanics using charm++. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium* (Hyderabad, India, May 2015), IEEE, pp. 677–686. <https://doi.org/10.1109/IPDPS.2015.45>.
- [NMK\*06] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. *Comput. Graph. Forum* 25, 4 (May 2006), 809–836.
- [NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.* 31, 6 (Nov. 2012). <https://doi.org/10.1145/2366145.2366171>.
- [PLPM20] 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* (Seattle, WA, USA, June 2020), IEEE, pp. 7363–7373. <https://doi.org/10.1109/CVPR42600.2020.00739>.
- [RPC\*10] ROHMER D., POPA T., CANI M.-P., HAHMANN S., SHEFFER A.: Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graph.* 29, 6 (Dec. 2010). <https://doi.org/10.1145/1882261.1866183>.
- [SGOC20] SANTESTEBAN I., GARCÉS E., OTADUY M. A., CASAS D.: SoftSMPL: Data-driven modeling of nonlinear soft-tissue dynamics for parametric humans. *Comput. Graph. Forum* 39, 2 (Oct. 2020), 65–75.
- [SLY20] SHI R., LI T., YAMAGUCHI Y.: Group visualization of class-discriminative features. *Neural Netw.* 129 (Sep. 2020), 75–90.
- [SOC19] SANTESTEBAN I., OTADUY M. A., CASAS D.: Learning-based animation of clothing for virtual try-on. *Comput. Graph. Forum* 38, 2 (Oct. 2019), 355–366.
- [SSIF09] SELLE A., SU J., IRVING G., FEDKIW R.: Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Trans. Vis. Comput. Graph.* 15, 2 (Nov. 2009), 339–350.
- [STOC21] SANTESTEBAN I., THUÉREY N., OTADUY M. A., CASAS D.: Self-supervised collision handling via generative 3D garment models for virtual try-on. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (July 2021), IEEE, pp. 11763–11773. <https://doi.org/10.1109/CVPR46437.2021.01159>.
- [TBTP20] 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. In *Computer Vision—ECCV 2020—16th European Conference* (Glasgow, UK, Aug. 2020), Springer, pp. 1–18. [https://doi.org/10.1007/978-3-030-58580-8\\_1](https://doi.org/10.1007/978-3-030-58580-8_1).
- [TWT\*16] TANG M., WANG H., TANG L., TONG R., MANOCHA D.: CAMA: Contact-aware matrix assembly with unified collision handling for GPU-based cloth simulation. *Comput. Graph. Forum* 35, 2 (Oct. 2016), 511–521.
- [VCC\*17] VELIČKOVIĆ P., CUCURULL G., CASANOVA A., ROMERO A., LIO P., BENGIO Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017). <http://arxiv.org/abs/1710.10903>.
- [VSGC20] VIDAURRE R., SANTESTEBAN I., GARCÉS E., CASAS D.: Fully convolutional graph neural networks for parametric virtual try-on. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Canada, Nov. 2020), Eurographics Association. <https://doi.org/10.1111/cgf.14109>.
- [WWYW20] WU L., WU B., YANG Y., WANG H.: A safe and fast response method for GPU-based cloth self collisions. *ACM Trans. Graph.* 40, 1 (Dec. 2020), 5:1–5:18.
- [XZK\*20] XU Z., ZHOU Y., KALOGERAKIS E., LANDRETH C., SINGH K.: RigNet: Neural rigging for articulated characters. *ACM Trans. Graph.* 39, 4 (Jan. 2020), 58:1–58:12.
- [ZWCM21] ZHANG M., WANG T. Y., CEYLAN D., MITRA N. J.: Deep detail enhancement for any garment. *Comput. Graph. Forum* 40, 2 (July 2021), 399–411.