# GarMatNet: A Learning-based Method for Predicting 3D Garment Mesh with Parameterized Materials

Zhen Luo
luo.utokyo@gmail.com
The University of Tokyo
Meguro, Tokyo, Japan

Tianxing Li
li-tianxing@hotmail.com
The University of Tokyo
Meguro, Tokyo, Japan

Takashi Kanai
kanait@acm.org
The University of Tokyo
Meguro, Tokyo, Japan

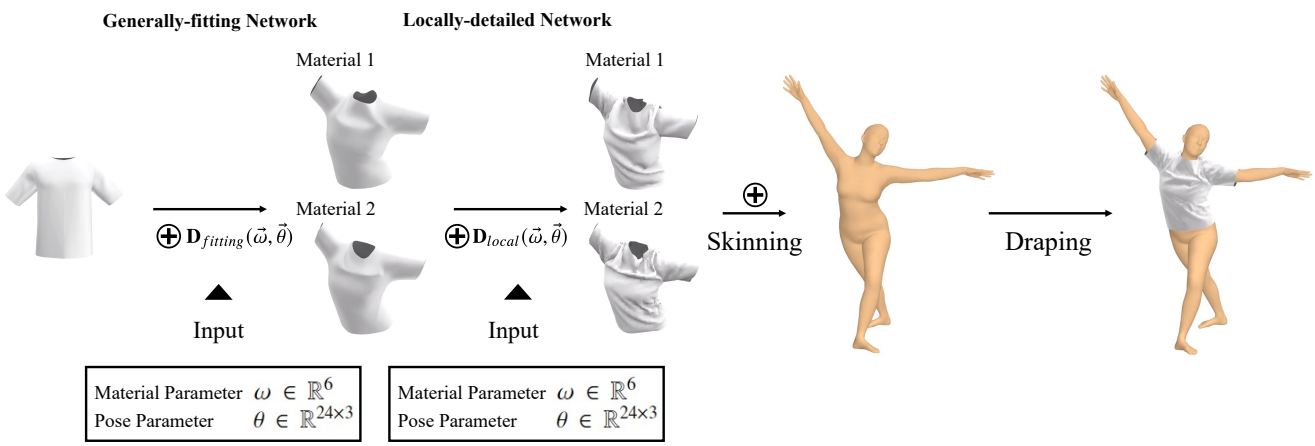


**Figure 1: Whole pipeline of GarMatNet.**

## ABSTRACT

Recent progress in learning-based methods of garment mesh generation is resulting in increased efficiency and maintenance of reality during the generation process. However, none of the previous works so far have focused on variations in material types based on a parameterized material parameter under static poses. In this work, we propose a learning-based method, GarMatNet, for predicting garment deformation based on the functions of human poses and garment materials while maintaining detailed garment wrinkles. GarMatNet consists of two components: a generally-fitting network for predicting smoothed garment mesh and a locally-detailed network for adding detailed wrinkles based on smoothed garment mesh. We hypothesize that material properties play an essential role in the deformation of garments. Since the influences of material type are relatively smaller than pose or body shape, we employ linear interpolation among different factors to control deformation. More specifically, we apply a parameterized material space based on the mass-spring model to express the difference between materials and construct a suitable network structure with weight adjustment between material properties and poses. The experimental results demonstrate that GarMatNet is comparable to the physically-based simulation (PBS) prediction and offers advantages regarding generalization ability, model size, and training time over the baseline model.


*MIG '21, November 10–12, 2021, Virtual Event, Switzerland*

ACM ISBN 978-1-4503-9131-3/21/11…$15.00
https://doi.org/10.1145/3487983.3488294

## CCS CONCEPTS

• **Computing methodologies** → **Animation**; *Artificial intelligence.*

## KEYWORDS

Neural networks, garment deformation, animation

**ACM Reference Format:**
Zhen Luo, Tianxing Li, and Takashi Kanai. 2021. GarMatNet: A Learning-based Method for Predicting 3D Garment Mesh with Parameterized Materials. In *Motion, Interaction and Games (MIG '21), November 10–12, 2021, Virtual Event, Switzerland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3487983.3488294

## 1 INTRODUCTION

Mesh generation of clothed humans benefits a lot of applications such as virtual try-on, game, digital fashion design, and 3D content production. With the increasing need for interactive applications, demands for real-time 3D visualization are increasingly growing. However, the simulation of garments on the human body is still

challenging in computer graphics research because of the complexity of garment wrinkles in various poses, body shapes, garment styles, and garment materials.

Previous works have two research directions. The first one, a predominant method, is Physically-Based Simulation (PBS), aiming to simulate complex clothing deformations based on physical equations. Research works such as [Baraff and Witkin 1998; Breen et al. 1994; Jiang et al. 2017; Nealen et al. 2006; Selle et al. 2009; Terzopoulos et al. 1987] successfully obtained high-quality results. However, two problems still exist in high-quality PBS methods: (a) High computational cost; It usually takes hundreds of hours to finish the computation. Even for a coarse garment mesh with 3,000 vertices, it still takes nearly 200 ms to compute each frame, making it almost impossible to satisfy the need for real-time simulation for daily applications. (b) Requirement for expert knowledge. The common pipeline of PBS methods usually includes editing the garment shape in 2D with patterns, manually placing the garment on the digital character, and setting up hyperparameters such as time step, material properties, etc., to achieve desired results. Both are laborious and need expert knowledge, e.g., in the field of fashion design or numerical simulation.

The second research direction is to build a real-time physically-aware learning-based method for generating 3D-clothed human animation. Recent works focus on learning approximate models from PBS compiled off-line. However,there seems to be no works exploring types of materials as a feature in the function of garment deformation under the static human body. However, the material has substantial effects on the deformation of clothed garments, and there are countless types of materials in the real world, making the problem even more challenging.

To tackle the above-mentioned material issue, we herein propose a learning-based approach, Garment Material Network (*GarMatNet*), to generate clothed garment mesh in different parameterized materials. Inspired by the previous PBS method [Provot 1995], namely, the mass-spring model, we describe the properties of materials by using three types of springs and one property of masses. In short, we leverage a 6-dimensional parameter to represent different materials, which covers $100^6$ types of materials. Also, TailorNet [Patel et al. 2020] shows that it is difficult for neural models to learn due to detailed wrinkles and the difference between materials. To further tackle the learning problem, we design GarMatNet with two components: a generally-fitting network to dress the clothes on the human body without detailed wrinkles and a locally-detailed wrinkle network to add detailed wrinkles to the mesh generated by the fitting network.

To be more specific, the input of GarMatNet consists of two parts: a 6-dimensional parameter representing parameterized materials and a 72-dimensional parameter representing the pose of the human body. The parameterized materials are based on the mass-spring method [Provot 1995]. The specific value of this 6-dimensional parameter for each type of material is determined by the normalized value from the library provided by a commercial 3D cloth design and simulation tool Marvelous Designer [CLO Virtual Fashion Inc. 2021], including 67 types of materials which are common to see in daily life. For pose parameter, we apply SMPL (A Skinned Multi-Person Linear Model) [Loper et al. 2015], a realistic 3D model of the human body that is based on skinning and blend shapes and is learned from thousands of 3D body scans. SMPL [Loper et al. 2015] provides a parameterized human body, including a pose and body shape. The output of GarMatNet is the deformation vector from smoothed garment to human body generated by the fitting network, and deformation vector from detailed-wrinkle garment to smoothed garment generated by the detailed-wrinkle network.

Our contributions are: (1) We propose the first learning-based approach as a joint model of parameterized garment's material and pose of the body, which is simple yet effective, and generalized well. (2) Compared to previous methods, our model has higher versatility, which can predict the deformation of garments made by different materials that the model has never seen before. (3) Compared to the PBS method, our model consumes less time and computational costs in realizing real-time simulation. (4) To accelerate further research, we will make publicly available a dataset with T-shirts made by 67 types of materials, simulated in 738 poses, totaling 49,446 samples.

## 2 RELATED WORK

The previous researches in the garment mesh generation field in the computer graphics community can be divided into two types: physically-based simulation (PBS) and learning-based methods. Physically-based garment simulation has been extensively used in current professional commercial software. However, due to the associated computational cost and stability concerns, the physically-based method is not suitable for real-time simulation. Accompanying the development of data science and machine learning, learning-based simulation has become increasingly popular and has the potential to be applied to real-time simulation.

### 2.1 Physically-based simulation

Physically-based simulation methods use discretizations of classical mechanics to deform the garment by solving an ordinary differential equation based on Newton's law [Nealen et al. 2006]. Based on this solution, several approaches by improving numerical methods, collision detection, and constraints have been proposed to increase realism, stability or to decrease the computational cost. However, due to the high computational cost and stability concern, the performance of the simulation is still limited in some ways.

The first physically-based approach developed in the long history of garment mesh generation is [Terzopoulos et al. 1987]. This approach is based on a continuum model and cannot express large deformations, and nonlinear constraints of garments. After that, many approaches have been studied, including the general particle system method [Baraff and Witkin 1998; Breen et al. 1994]. These methods can express real-world physical phenomena. However, they lack efficiency due to the need for high resolution or time step size.

Therefore, one of the primary limitations of physically-based methods of garment mesh generation is that with higher realism, higher computational cost is needed. They usually require significant run-time and make it impossible to apply these methods to real-time interactive applications. For example, a super-realistic animation requires a resolution of millions of triangles [Jiang et al. 2017; Selle et al. 2009; Terzopoulos et al. 1987], which is computationally expensive. A lot of researches focus on increasing the efficiency of the simulation. For example, the position-based method

[Müller et al. 2007] can produce approximated but acceptable results. However, it may lack some realism that cannot be applied for real-world applications such as fashion design and virtual try-ons. In addition, projecting the complicated equations of motion into a simpler subspace [de Aguiar et al. 2010], or subspace method, has also been proven effective in increasing the computational efficiency. However, the universality of this method is narrow and can only be used for specific scenes.

Besides the above methods, a simpler physically-based model, such as mass-spring model [Provot 1995] has been proposed. Provot builds a mass-spring garment model based on a regular quadrilateral mesh grid. In this work, the relation between each mass point could be described by three types of springs. The mass-spring model simplified the complicated physical properties of the garment, and the computational efficiency is extremely higher than the general particle system method, but it still has a problem. This method can only be applied to the rectangular mesh, which is different from the triangle mesh commonly used in the Computer Graphics community. To solve this problem, [Baraff and Witkin 1998] introduced a mass-spring model for a triangular mesh to expand the scope of application of the mass-spring model. In addition, some other works also focus on increasing the efficiency of PBS method, such as [Fratarcangeli et al. 2016; Wang and Yang 2016].

## 2.2 Learning-based simulation method

With the recent success of deep learning methods in various imaging and 3D geometry task, the current research trend is to learn garment deformation under body motion by applying neural networks.

Recent researchers [Casas and Otaduy 2018; Lewis et al. 2000] propose to predict deformation as a function of human poses and shapes. By combining this above above method and parametric human body models with per-vertex displacement, other works [Alldieck et al. 2019; Bhatnagar et al. 2019] successfully increase the reality of the results. Although these methods have better efficiency compared to the physically-based simulation method, they only work well for tight clothes such as t-shirt or pants which are fitted closely to the body surface without some complicated wrinkles.

Gundogdu et al. [Gundogdu et al. 2019] build GarNet, a three-step deep neural network to predict the deformation of garments by extracting garment features at varying levels of detail, including point-wise, patch-wise, and global features. Santesteban et al. [Santesteban et al. 2019] also present a learning-based method for clothing animation based on a database generated by PBS. Their approach learns coarse garment shapes based on the body shape and detailed wrinkles based on pose dynamics. Although their work can predict garments made of different materials, it requires independent training per garment material. More recently, Patel et al. [Patel et al. 2020] have proposed an MLP-based neural network model to predict garment deformation in 3D as a function of three components: pose, shape, and style (garment geometry), while maintaining detailed wrinkle. In their work, they decomposed the complicated deformation of the garment into high-frequency and low-frequency components. They predict the low-frequency component from the pose, body shape, and style parameters with an MLP and predict the high-frequency component by a mixture of 20 shape-style specific pose models. Although their work is the first approach to extend the universality of the model to cover pose, body shape, and garment style, the method still does not deal with different garment materials. Zhang et al. [Zhang et al. 2021] enhance, in a data-driven manner, rich yet plausible details starting from a coarse garment geometry by applying a normal map of garment mesh as a feature in their training network. Although their work can deal with the prediction of garment structure between different materials, their approach can only transfer the materials only if the database generated by these materials has been trained in the network, due to the lack of generalization.

Looking through the history of the learning-based method, we can see the primary trend in this research field includes: implementation of a novel deep learning model, expanding the scope of application, as well as increasing more valuable information as features. Most of the learning-based methods have higher efficiency in generating garment mesh compared with the physically-based simulation method, which means that it has strong potential to be applied to real-time animation generation. In our work, we focus on the learning-based method to ensure higher efficiency and lower requirement of hardware. We also focus on expanding the scope of application to parameterized materials.

# 3 METHOD

## 3.1 Material property space

There are several models capable of describing the properties of materials such as [Clyde et al. 2017; Miguel et al. 2012; Wang et al. 2011]. At here, we leverage a material property space inspired by the mass-spring model [Provot 1995]. The mass-spring model is a physically-based model for animating cloth objects. Let the elastic model of a garment be a mesh of $m \times n$ virtual masses, and each mass is linked to its neighbors by several weightless springs of natural length non-equal to zero. The linkage between two masses can be described by the following three types in Figure 2:

(1) springs linking masses $[i, j]$ and $[i+1, j]$, and masses $[i, j]$ and $[i, j+1]$, are called as "structural springs", to describe the force between two adjacent masses in vertical and horizontal directions;
(2) springs linking masses $[i, j]$ and $[i+1, j+1]$, and masses $[i+1, j]$ and $[i, j+1]$, are called as "shear springs", in order to describe the force between two adjacent masses in diagonal direction;
(3) springs linking masses $[i, j]$ and $[i+2, j]$, and masses $[i+1, j]$ and $[i, j+2]$, are called as "flexion springs", in order to describe the force between two distant masses in the vertical and horizontal directions.

Using these three types of springs, and the weight of each mass (usually seen as density), we can describe the physical properties of any material. We conduct our simulation based on the mass-spring model in Marvelous Designer [CLO Virtual Fashion Inc. 2021], and change the parameter of materials by changing the elastic coefficient of the spring in the mass-spring model.

We have a two-dimensional parameter to describe the strength of structural springs in vertical and horizontal directions separately, a two-dimensional parameter to describe the strength of flexion

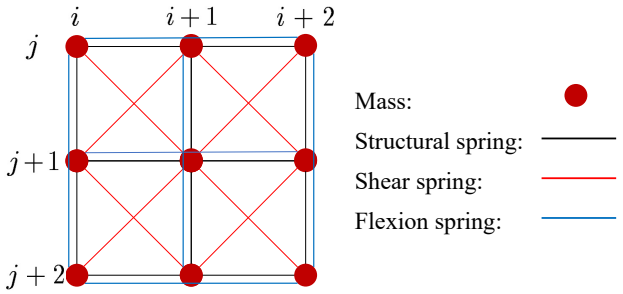


**Figure 2: Mesh structure of mass-spring model.**

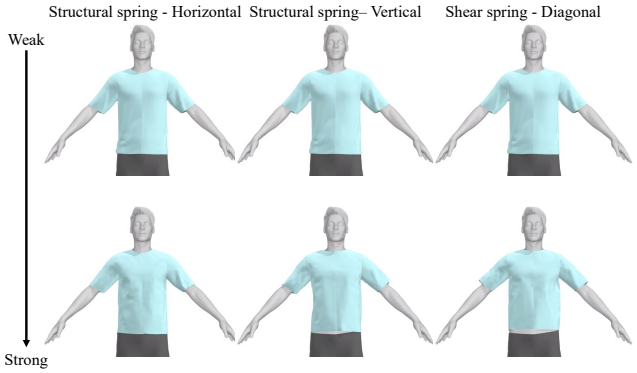


**Figure 3: Sensitivity test of structural spring and shear spring.**

springs in the vertical and horizontal directions separately, a dimensional parameter to describe the strength of shear springs in the diagonal direction, and a dimensional parameter to describe the mass or density of the target material. In our method, we denote the material parameter as $\vec{\omega}$, which consists of those six-dimensional parameters. The value of $\vec{\omega}$ for each type of material is decided by normalized value from the library provided by Marvelous Designer [CLO Virtual Fashion Inc. 2021].

To validate the mass-spring model, we conduct the sensitivity tests under the same pose with different spring strengths. The results of structural spring and shear spring are shown in Figure 3, while the results of flexion spring and density are shown in Figure 4. From Figure 3, we found that when the strength of structural springs is increased in the horizontal and vertical directions, the garment becomes more rigid in its corresponding direction, and detailed wrinkles are generated due to the change in the strength of the springs. This change in the garment more like the difference in stiffness between soft cotton and hard paper. Also, when shear springs' strength increases, the garment becomes stiffer in the diagonal direction and looks more supportive.

From Figure 4, the change is slightly different from the previous two types of springs. For flexion spring, it can be seen that with increasing strength, the material becomes harder, resembling the change in stiffness from the softness of cotton to the hardness of

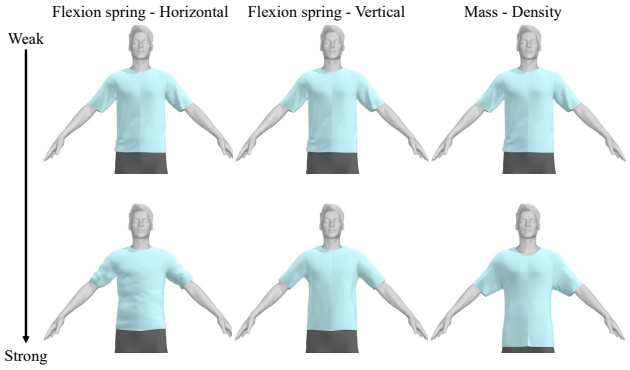


**Figure 4: Sensitivity test of flexion spring and density.**

plastic, and the number of wrinkles decreases. Lastly, the effects generated by density are easy to understand. When the material density is increased, it gives us a feeling of sagging, similar to the skin. Also, when the density becomes extremely large, the stretch deformation of the garment occurs, leading to the increase in the length of the original garment.

## 3.2 GarMatNet

Inspired by TailorNet [Patel et al. 2020], we also divide our prediction model into two parts, a generally-fitting network, and a locally detailed network. The generally-fitting network is installed to decrease the task's difficulty: we firstly predict the displacement between the human body and a smoothed garment generated by data processing based on the simulated result. Compared with the simulated result, a smoothed garment has a more straightforward structure and fewer small-scale wrinkles, which means that it is easy to be predicted. Generally-fitting network is designed based on MLP (Multi-Layer Perceptron) and used for predicting the displacement between a smoothed garment and human body $\mathbf{D}_{fitting} \in \mathbb{R}^{n\times 3}$, as a function of pose $\vec{\theta}$ and type of garment material $\vec{\omega}$. The locally-detailed network is set for predicting the displacement between a smoothed garment and simulated garment $\mathbf{D}_{local} \in \mathbb{R}^{n\times 3}$, also as a function of pose $\vec{\theta}$ and type of garment material $\vec{\omega}$. In total, we have a two-step prediction model to predict the displacement between a simulated garment and human body $\mathbf{D}_{whole}$ by Equation 1. By adding the displacement to the human body, we finally obtain the mesh of a garment made by the target material under the corresponding pose. The detailed information about our prediction model is described in Figure 1.

$$\mathbf{D}_{whole}(\vec{\omega}, \vec{\theta}) = \mathbf{D}_{fitting}(\vec{\omega}, \vec{\theta}) + \mathbf{D}_{local}(\vec{\omega}, \vec{\theta}). \tag{1}$$

Compared to previous researches from Table 1, GarMatNet has higher versatility than other previous methods by material variations. This method can produce garments made of different materials, and the number of materials that we can deal with is unlimited. In addition, we use PBS data as our ground truth, which means our dataset is easy to expand and costs less than real-captured data.

| Method | Pose Variations | Material Variation | PBS data |
|---|---|---|---|
| Wang et al., 2018 [Wang et al. 2018] | No | No | Yes |
| TailorNet 2020 [Patel et al. 2020] | Yes | No | Yes |
| CAPE 2020 [Ma et al. 2020] | Yes | No | No |
| GarMatNet 2021 | Yes | Yes | Yes |

**Table 1: Comparison between GarMatNet and previous researches.**

## 3.3 Decomposing the cloth displacement

Inspired by TailorNet [Patel et al. 2020], we hypothesize that it is hard to directly predict the distance between a simulated garment and human body $\mathbf{D}(\boldsymbol{\theta}, \omega)$ with MLP. Our experiments show that this pipeline would make the model hard to learn, and the prediction results look unreal and lack details. Therefore, we divide our model into two parts. Generally-fitting network is designed based on MLP (Multi-Layer Perceptron). It is used for predicting the displacement between a smoothed garment and human body $\mathbf{D}_{fitting}$ as a function of pose $\theta$ and type of garment material $\vec{\omega}$. The locally-detailed network is set for predicting the displacement between a smoothed garment and simulated garment $\mathbf{D}_{local}$, also as a function of pose $\vec{\theta}$ and type of garment material $\vec{\omega}$. To create the label for the training of a generally-fitting network, it is necessary to decompose the cloth displacement.

In order to decompose the cloth displacement, we use the garment template topology $\mathbf{T}^G$, and process $\mathbf{T}^G$ by Laplacian smoothing [Vollmer et al. 1999]. With Laplacian smoothing of $\mathbf{T}^G$, we are able to get a new smoothed garment template topology $\mathbf{T}^G_{fitting}$. Let $f(\mathbf{x}, t) : \mathcal{G} \mapsto \mathbb{R}$ be a function on the garment surface, then it is smoothed with the diffusion equation:

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = \lambda \Delta f(\mathbf{x}, t), \tag{2}$$

which means that the function changes over time by a scalar diffusion coefficient $\lambda$ times its spatial Laplacian $\Delta f$. We apply this Laplacian smoothing method to the vertex coordinates $\mathbf{t}_i \in \mathbf{T}^G$:

$$\mathbf{t}_i = \mathbf{t}_i + \lambda \Delta \mathbf{t}_i, \tag{3}$$

where $\Delta \mathbf{t}_i$ denotes the discrete Laplace-Beltrami operator applied at vertex $\mathbf{t}_i$, and $\lambda$ and the number of iterations controls the level of smoothing. A lower value of $\lambda$ yields a smoother surface and fewer structural details. We use $\lambda = 0.15$ and 80 iterations to obtain a smoothed garment topology $\mathbf{T}^G_{fitting}$. In total, we could achieve the labels for two networks in GarMatNet by:

$$\mathbf{D}_{local}(\vec{\omega}, \vec{\theta}) = \mathbf{T}^G(\vec{\omega}, \vec{\theta}) - \mathbf{T}^G_{fitting}(\vec{\omega}, \vec{\theta}), \tag{4}$$

$$\mathbf{D}_{fitting}(\vec{\omega}, \vec{\theta}) = \mathbf{D}(\vec{\omega}, \vec{\theta}) - \mathbf{D}_{local}(\vec{\omega}, \vec{\theta}). \tag{5}$$

## 3.4 Generally-fitting Network

Generally-fitting Network is used for predicting the displacement between body surface $M(\vec{\theta})$ and smoothed garment template topology $\mathbf{D}_{fitting}(\vec{\omega}, \vec{\theta})$, generated by Laplacian smoothing.

We implement a generally-fitting network with a three-layers neural network. The input of our network is pose parameter $\vec{\theta} \in \mathbb{R}^{24\times3}$ and material parameter $\vec{\omega} \in \mathbb{R}^6$. The output is the predicted displacement between human body $M(\vec{\theta})$ and a smoothed garment $\mathbf{D}_{fitting}(\vec{\omega}, \vec{\theta}) \in \mathbb{R}^{n\times3}$. We use Mean Square Error (MSE) as our loss function:

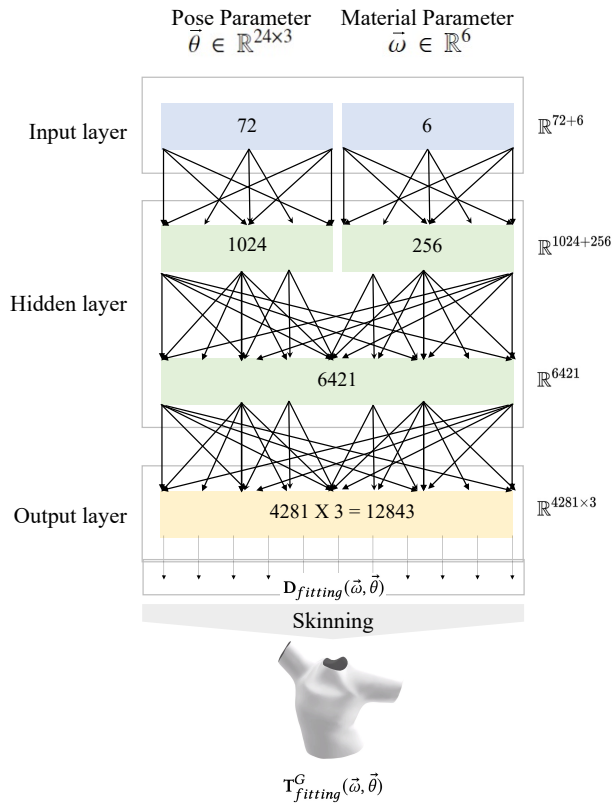


**Figure 5: Detailed network architecture of generally-fitting network.**

$$\text{Loss}_{fitting} = \frac{1}{n} \sum_{i=1}^{n} \left| \mathbf{D}_{fitting} - \mathbf{N}_{fitting}(\vec{\omega}, \vec{\theta}) \right|^2, \tag{6}$$

where $\mathbf{N}_{fitting}(\vec{\omega}, \vec{\theta})$ denotes the predicted displacement output by our generally-fitting network. In total, our loss function represents the average of the square error of each vertex. Figure 5 shows the detailed architecture of the generally-fitting network of GarMatNet.

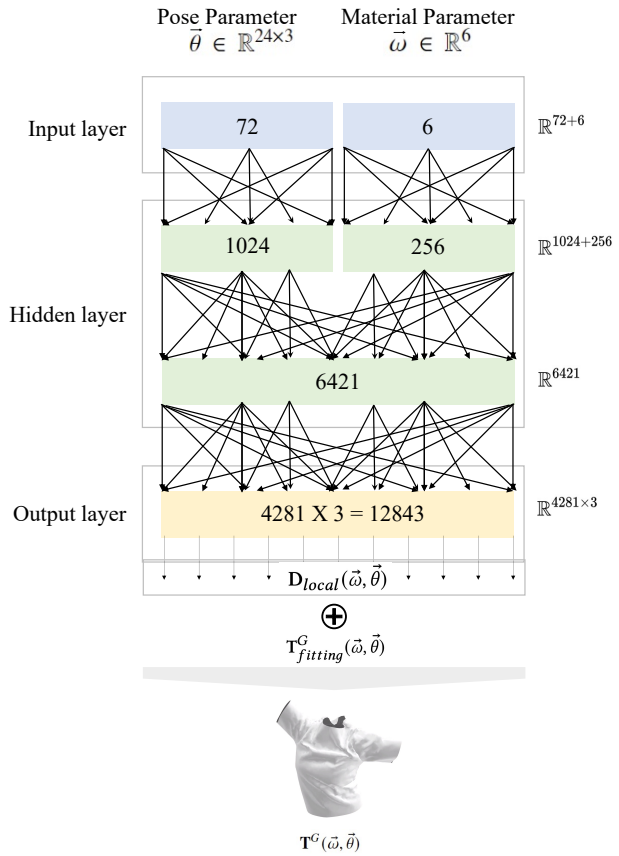


**Figure 6: Detailed network architecture of locally-detailed network.**

## 3.5 Locally-detailed Network

Locally-detailed Network is used for predicting the displacement between a smoothed garment topology $\mathbf{T}^{G}_{fitting}(\vec{\omega},\vec{\theta})$, and simulated garment topology $\mathbf{T}^{G}(\vec{\omega},\vec{\theta})$.

As well as the generally-fitting network, we also implement a locally-detailed network with the same three-layers neural network. The input is a pose parameter $\vec{\theta} \in \mathbb{R}^{24\times3}$ and a material parameter $\vec{\omega} \in \mathbb{R}^{6}$. The output is the predicted displacement between a smoothed garment topology and simulated garment topology $\mathbf{D}_{local}(\vec{\omega},\vec{\theta}) \in \mathbb{R}^{n\times3}$. We use Mean Square Error (MSE) as our loss function:

$$\text{Loss}_{local} = \frac{1}{n}\sum_{i=1}^{n}\left|\mathbf{D}_{local} - \mathbf{N}_{local}(\vec{\omega},\vec{\theta})\right|^{2}, \tag{7}$$

where $\mathbf{N}_{local}(\vec{\omega},\vec{\theta})$ denotes the predicted displacement output by our locally-detailed network, and $n$ (= 4281) be the number of vertices of garment mesh. In total, our loss function represents the average of the square error of each vertex. The detailed architecture of locally-detailed network of GarMatNet is shown in Figure 6.

Compared to previous works like TailorNet [Patel et al. 2020], architectures of the above two networks have the following differences: (1) We split pose parameter and material parameter at the input layer and the first hidden layer. The benefit of this structure is that it helps to input two parameters with different units into one network. (2) We use an early stopping method to decide when we should stop training instead of setting a fixed number of epochs. This design prevents over-fitting. (3) We use He Kaiming initialization [He et al. 2015] to enhance the efficiency of training process. (4) We normalize our output and change the activation function from ReLu to tanh.

## 3.6 Weight controller

We define a variable $\alpha$ to control the impact that the material properties have on the performance. The $\alpha$ is a real value between 0 and 1. The higher the $\alpha$ is, the more substantial impact it has during training and vice versa. We here add $\alpha$ to the material parameter $\vec{\omega}$ at the input layer and $1-\alpha$ to the pose parameter $\vec{\theta}$ at the input layer. The weight controller could help decide which part plays a more critical role in the deformation of garment and increase the accuracy of the prediction model.

## 3.7 Dataset

We build a dataset with 49446 instances to train and test our GarMatNet. We have 738 different static poses and 67 different types of materials which are commonly seen in daily life. The detailed dataset, or simulated results, are generated by PBS conducted in Marvelous Designer [CLO Virtual Fashion Inc. 2021]. For pose variations, we use 738 static SMPL poses, including a wide range of poses generated from SMPL public sample series. For a given type of material $\vec{\omega}_i$ and poses $\vec{\theta}_i$, we simulate them in a sequence. To avoid dynamic effects since we conduct the simulation by a pose series and it could generate some dynamic effects, we stop each pose and keep the body static for five frames to let the garment relax.

# 4 RESULTS

In this section, we discuss our experiments and obtained results. We evaluate our method on T-Shirt meshes quantitatively and qualitatively. Then we compare it to our baseline. Our baseline is designed based on the architecture of TailorNet [Patel et al. 2020] by discarding our parameterized material space $\vec{\omega}$.

We perform experiments on Intel(R) Xeon(R) W-2123 @3.60GHz CPU and NVIDIA GeForce RTX 2080Ti GPU. We evaluate our method mainly by comparing our results with a baseline and the ground truth generated by PBS. For training, Pytorch [Paszke et al. 2019] is used for coding and construct the machine learning model.

## 4.1 Experimental setting

Our baseline and GarMatNet use several MLPs - each of them has two hidden layers with tanh activation and a dropout layer. We obtain optimal hyperparameters by tuning our baseline and then keep them constant to train all other MLPs. We use early stopping, a form of regularization for preventing overfitting when training a learner with an iterative method. The value of patience for the

| Method | Network | Error(mm) |
|---|---|---|
| GarMatNet | Generally-fitting network | 3.10 |
| | Locally-detailed network | 2.58 |
| | Whole pipeline | 5.47 |
| Baseline (TailorNet) | Generally-fitting network | 5.68 |
| | Locally-detailed network | 3.09 |
| | Whole pipeline | 8.45 |

**Table 2: Mean per-vertex error of all testing instances.**

generally-fitting network is 50 and for the locally-detailed network is 200 respectively.

For training and testing GarMatNet, we split our dataset into three subsets: (1) test set: including four different types of materials selected randomly with 73 poses. All of the data related to these 73 poses are moved to test to make sure that the model would never meet the information of these target poses. Also, all of the remaining poses in these four test materials would be moved to test to ensure that the model never meets any information about the four test materials. In total, there are four types of unseen materials with 73 unseen poses. (2) training set: 90% of data except for test. (3) development set: remaining 10% of data except for test.

For the baseline, since we train a network for a baseline separately, we select the same four types of materials as our target materials and use their data in training. For the target material, we would firstly select 73 poses mentioned above as the test pose, and we replicate the remaining data 67 times to keep the number of instances close to the training data used for GarMatNet's training. Dataset is used by training, test, and development at the same rate as the case of GarMatNet

To test the effectiveness of GarMatNet, we select four types of materials and 73 poses, and our data splitting method is based on making these poses and materials are unseen. Our four target types of materials are Sherpa Fleece 160, silk Duchess Satin, Trim fusible rigid, and cashmere. All of these materials are common in daily life and have different properties which can also be distinguished visually. Our target poses are selected from a 738-frame dancing animation and their frame number is divisible by 10.

## 4.2 Experimental results

Firstly, we conduct experiments with GarMatNet and baseline model based on TailorNet [Patel et al. 2020]. We select the mean distance error of all vertices, including the mean per-vertex error of all test instances and that of each target pose and material.

The mean distance error of all test samples is shown in Table 2. We found that our GarMatNet generally outperforms the baseline in both networks and the whole pipeline. This is that, because of the application of parameterized material, GarMatNet can study the difference amongst different poses from data on other materials'. On the other hand, the baseline model can only learn the pose information from the data of the material it uses, which means that the data cannot be leveraged for other garments.

In addition, the mean per-vertex error of each test pose in each test material is shown in Figure 7 for sherpa fleece and silk, and in Figure 8. Compared with the baseline method, GarMatNet performs

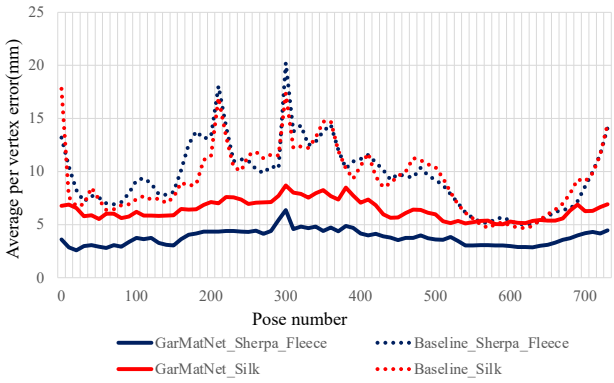


**Figure 7: Mean per-vertex error of each test pose in each test material (sherpa fleece and silk).**

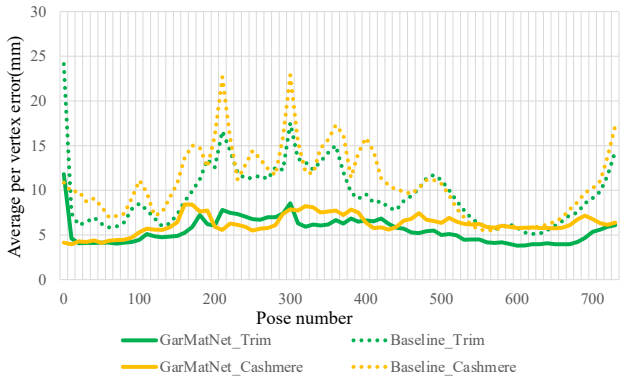


**Figure 8: Mean per-vertex error of each test pose in each test material (trim and cashmere).**

better in most cases. Also, we found that when the value of mean per-vertex error is not the same at every pose, especially a pose at the 300th frame, there is a peak in error. We consider that the data distribution is not balanced, and the degree of pose change around the 300th frame is more significant than the other poses, and it is the reason for the unequal error.

We also visually compare the results of GarMatNet and the baseline model. Since our task is to learn the differences between different types of materials and generate suitable wrinkles on the posed human body, it is necessary to evaluate whether our model can retain detailed information about wrinkles depending on the type of material.

We firstly illustrate the rendered images of ground truth data, predictions by GarMatNet, and predictions by baseline model. And then, we show the error between ground truth and predicted results by heat map to visualize the error information. Moreover, we also use SSIM (Structural Similarity) [Wang et al. 2004] to evaluate the similarity between the ground truth and the predicted result quantitatively.

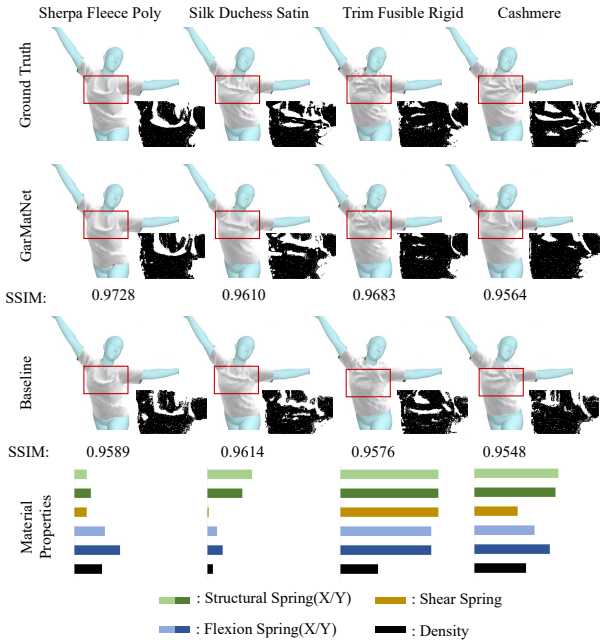


**Figure 9: Rendered results of Pose A.**

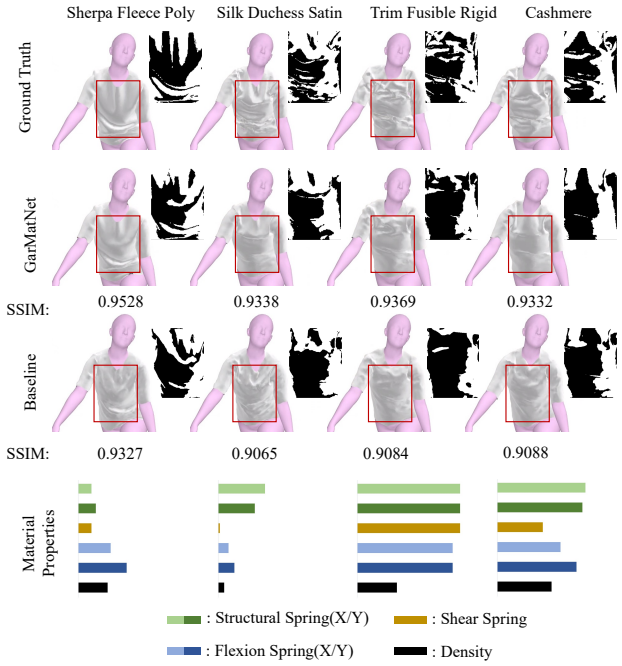


**Figure 10: Rendered results of Pose B.**

We select pose A, B corresponding to 600th and 730th frame and evaluate their rendered results. The results of poses A and B are shown in Figure 9 and Figure 10 respectively. Apart from the rendered results and SSIM we mentioned before, we also add the black-white local enlargement of wrinkle details to provide a clearer description of the texture of the wrinkles. Also, we show the value of each dimension in our parameterized material at the bottom of each figure.

We can observe that by SSIM, our GarMatNet method outperforms the baseline method in most cases. Although for silk in pose A, GarMatNet performs slightly poorer than baseline. However, GarMatNet's results are better than baseline in most cases. The high value of SSIM proves that GarMatNet is able to retain fine details and generate a mesh with suitable topology in most cases. For silk and cashmere cases, we observe that the performance is poorer than the other two materials. Also, in pose B, the SSIM of GarMatNet seems lower than the other pose. We consider the reasons for these problems to be the imbalance of our dataset.

From the black-white local enlargements of wrinkles' detail, we observe that apparently, the textures of wrinkles are different depending on the types of material. In addition, GarMatNet's black-white enlargements have roughly the same structure as ground truth's in most cases. We realize that in some cases, such as Sherpa Fleece in Pose B, the baseline's black-white enlargement is not able to show the same wrinkle's pattern as ground truth's, but GarMatNet predicts a similar pattern like the ground truth one. This can also indicate that qualitatively, GarMatNet is doing better than baseline.

Moreover, by considering the value of each dimension in our parameterized material model, we can use our intuitive sense of vision to evaluate our approach. We can observe that when the structural springs and flexion springs are weak, like sherpa fleece, the number of wrinkles is lower than the other materials, and the characteristics of these wrinkles are soft and large-scale. In addition, when all the springs are strong, like trim, there will be more wrinkles, and the garment would look more supportive and as hard as plastics, but retain the "crisp" characteristic of paper at the same time.

Lastly, we compare GarMatNet results and baseline results with the ground truth by heat map to evaluate our method locally. The results of pose A and pose B are shown in Figure 11 and in Figure 12 respectively. The heat map is leveraged here to evaluate the performance of GarMatNet at each location on the garment. The colors stand for the distance error between each pair of vertices. From the above heat maps, we found that GarMatNet performs well in these three poses and better than baseline in most cases. It is easy for the baseline method to generate a significant error at sleeves, while GarMatNet can suppress large errors at the sleeves. Also, for pose B, the baseline's prediction is slightly far from the ground truth, especially near the neckline. However, GarMatNet can deal with these cases will and predict the mesh structure with reality.

## 4.3 Weight controller

The weight controller is applied at the input layer as $\alpha$. We conduct experiments by GarMatNet with $\alpha$= 0.1, 0.3, 0.5, 0.7 and 0.9. The higher the value of $\alpha$, the higher the weight of material properties in the network. The results are shown in Table 3. We can observe that by changing the weight manually, and we can adjust

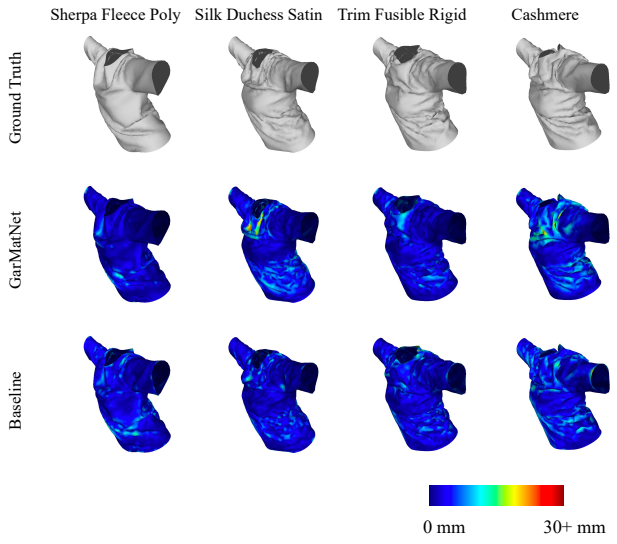


**Figure 11: Heat map of results of Pose A.**

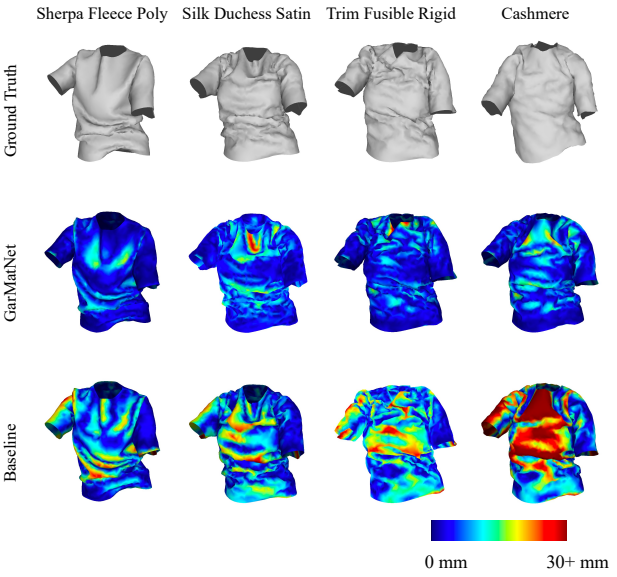


**Figure 12: Heat map of results of Pose B.**

the accuracy of GarMatNet. By our manual experiments, we select $\alpha = 0.5$ as the general setting for all experiments in this paper.

## 4.4 Training time and performance

The total training time for GarMatNet is about 14 hours, and for baseline, it is four hours for a network, a total of 16 hours for four types of materials. The model size of GarMatNet is around 450MB, while for baseline is 350MB for a network, a total of 1800MB for four types of materials. The prediction time of GarMatNet and baseline method is about 20ms/frame. We utilize PBS to generate our dataset, and the computational time for the PBS method is 325ms/frame by an NVIDIA GeForce GTX 1060Ti GPU. In total, GarMatNet performs well in terms of computational costs and meets the requirement for generating real-time animation with 30fps, while PBS cannot.

| $\alpha$ | Generally-fitting network(mm) | Locally-detailed network(mm) | Whole pipeline(mm) |
|---|---|---|---|
| 0.1 | 3.10 | 3.38 | 6.01 |
| 0.3 | 3.13 | 2.68 | 5.56 |
| 0.5 | 3.10 | 2.58 | 5.47 |
| 0.7 | 3.98 | 2.91 | 6.12 |
| 0.9 | 4.24 | 2.98 | 6.45 |

**Table 3: Mean per vertex error of GarMatNet under different weights.**

Compared to the baseline method, GarMatNet has better generalization capability in terms of the ability to predict garments' meshes made for an unlimited number of materials. Moreover, it also has better performance in terms of training time, error, model size, and visual performance.

# 5 CONCLUSIONS AND FUTURE WORK

In this paper, we introduce GarMatNet, a learning-based approach for predicting the mesh structure of garments made by different materials on the different posed human body. Our idea is to leverage a parameterized material model inspired by the mass-spring model and construct a suitable MLP-based network for predicting the subtle difference between different materials. GarMatNet consists of two parts: a generally-fitting network for predicting smoothed garments and a locally-detailed network for generating detailed wrinkles on a garment. Our experiments show that both these two networks outperform in terms of mean per-vertex error and visualization. GarMatNet also has a strong generalization capacity because of the application of parameterized material. It can be used for predicting garments made of any material not found in the training. We also compare our method with a baseline based on the structure of TailorNet [Patel et al. 2020]. The results show that we can obtain better results quantitatively and qualitatively with a shorter training time and smaller model size.

While the GarMatNet method shows impressive generalization capability, it has several limitations that can be addressed in future work. First, GarMatNet cannot deal with human bodies with different body shapes and garment styles. This work needs a considerable number of data that exceeds the capability of current mainstream hardware. Therefore, it is important to develop an appropriate and efficient way to generate data and construct a suitable network to train it with a limited number of data. Secondly, more features can be added to the model, and more accurate results should be obtainable -with the use of effective features such as a normal map. Thirdly, some properties related to the material are not considered in this work, such as the frictional force between garment and skin. Leveraging a more exquisite model to describe material properties may help contribute to increasing the reality.

## ACKNOWLEDGMENTS

We gratefully thank Dr. Qiqi Gao (AI lab, CyberAgent,Inc.) for her suggestions on mesh processing and Dr. Yang Zhao (IBM Research) for his suggestions on deep learning. Tianxing Li acknowledges the receipt of Japanese government (MEXT) scholarships. This work was partially supported by JSPS KAKENHI, Grant Number JP19K11990.

## REFERENCES

Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019. Learning to Reconstruct People in Clothing From a Single RGB Camera. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1175–1186. https://doi.org/10.1109/CVPR.2019.00127

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54. https://doi.org/10.1145/280814.280821

Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. 2019. Multi-Garment Net: Learning to Dress 3D People From Images. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 5419–5429. https://doi.org/10.1109/ICCV.2019.00552

David E. Breen, Donald H. House, and Michael J. Wozny. 1994. Predicting the Drape of Woven Cloth Using Interacting Particles. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. Association for Computing Machinery, New York, NY, USA, 365–372. https://doi.org/10.1145/192161.192259

Dan Casas and Miguel A. Otaduy. 2018. Learning Nonlinear Soft-Tissue Dynamics for Interactive Avatars. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1, Article 10 (July 2018), 15 pages. https://doi.org/10.1145/3203187

CLO Virtual Fashion Inc. 2021. *Marvelous Designer*. https://www.marvelousdesigner.com/ Accessed June, 2021.

David Clyde, Joseph Teran, and Rasmus Tamstorf. 2017. Modeling and Data-Driven Parameter Estimation for Woven Fabrics. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 17, 11 pages. https://doi.org/10.1145/3099564.3099577

Edilson de Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K. Hodgins. 2010. Stable Spaces for Real-Time Clothing. *ACM Trans. Graph.* 29, 4, Article 106 (July 2010), 9 pages. https://doi.org/10.1145/1778765.1778843

Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-Seidel Method for Stable Soft Body Dynamics. *ACM Trans. Graph.* 35, 6, Article 214 (Nov. 2016), 9 pages. https://doi.org/10.1145/2980179.2982437

Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. 2019. GarNet: A Two-Stream Network for Fast and Accurate 3D Cloth Draping. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 8738–8747. https://doi.org/10.1109/ICCV.2019.00883

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 1026–1034. https://doi.org/10.1109/ICCV.2015.123

Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. Graph.* 36, 4, Article 152 (July 2017), 14 pages. https://doi.org/10.1145/3072959.3073623

J. P. Lewis, Matt Cordner, and Nickson Fong. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 165–172. https://doi.org/10.1145/344779.344862

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graph.* 34, 6, Article 248 (Oct. 2015), 16 pages. https://doi.org/10.1145/2816795.2818013

Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. 2020. Learning to Dress 3D People in Generative Clothing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 6468–6477. https://doi.org/10.1109/CVPR42600.2020.00650

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. http://www.gmrv.es/Publications/2012/MBTBMOM12

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118. https://doi.org/10.1016/j.jvcir.2007.01.005

Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. 2006. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum* 25, 4 (2006), 809–836. https://doi.org/10.1111/j.1467-8659.2006.01000.x

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. 2020. TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7363–7373. https://doi.org/10.1109/CVPR42600.2020.00739

Xavier Provot. 1995. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour. In *Proceedings of Graphics Interface '95* (Quebec, Quebec, Canada) *(GI '95)*. Canadian Human-Computer Communications Society, Toronto, Ontario, Canada, 147–154. http://graphicsinterface.org/wp-content/uploads/gi1995-17.pdf

Igor Santesteban, Miguel A. Otaduy, and Dan Casas. 2019. Learning-Based Animation of Clothing for Virtual Try-On. *Computer Graphics Forum* 38, 2 (2019), 355–366. https://doi.org/10.1111/cgf.13643

Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. 2009. Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction. *IEEE Transactions on Visualization and Computer Graphics* 15, 2 (March 2009), 339–350. https://doi.org/10.1109/TVCG.2008.79

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. Association for Computing Machinery, New York, NY, USA, 205–214. https://doi.org/10.1145/37401.37427

J. Vollmer, R. Mencl, and H. Müller. 1999. Improved Laplacian Smoothing of Noisy Surface Meshes. *Computer Graphics Forum* 18, 3 (1999), 131–138.

Huamin Wang, Ravi Ramamoorthi, and James F. O'Brien. 2011. Data-Driven Elastic Models for Cloth: Modeling and Measurement. *ACM Transactions on Graphics* 30, 4 (July 2011), 71:1–11. http://graphics.berkeley.edu/papers/Wang-DDE-2011-08/ Proceedings of ACM SIGGRAPH 2011, Vancouver, BC Canada.

Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Trans. Graph.* 35, 6, Article 212 (Nov. 2016), 10 pages. https://doi.org/10.1145/2980179.2980236

Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popović, and Niloy J. Mitra. 2018. Learning a Shared Shape Space for Multimodal Garment Design. *ACM Trans. Graph.* 37, 6, Article 203 (Dec. 2018), 13 pages. https://doi.org/10.1145/3272127.3275074

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

Meng Zhang, Tuanfeng Y. Wang, Duygu Ceylan, and Niloy J. Mitra. 2021. Deep Detail Enhancement for Any Garment. *Computer Graphics Forum* 40, 2 (2021), 399–411. https://doi.org/10.1111/cgf.142642