

# Hybrid Learning with Multi-Scale Graphs for Enhanced Garment Deformation Approximation

Tianxing Li<sup>a</sup>, Rui Shi<sup>b</sup>, Qing Zhu<sup>a</sup>, Takashi Kanai<sup>c</sup>

<sup>a</sup>*College of Computer Science, Beijing University of Technology, Beijing, 100124, China*

<sup>b</sup>*School of Information Science and Technology, Beijing University of Technology, Beijing, 100124, China*

<sup>c</sup>*Department of General Systems Studies, University of Tokyo, Tokyo, 153-8902, Japan*

---

## Abstract

Due to the complex behavior of clothing, modelling fine-scale garment deformation on arbitrary meshes within a unified network presents a considerable challenge. Current methods often fail to learn generalized patterns that adhere to physical laws in a controllable manner, compromising efficiency and realism in practical applications. To overcome these limitations, we introduce a novel hybrid learning approach that accurately simulates garment dynamics and intricate details. The core of our method is a progressive learning scheme that integrates the accuracy of supervised learning with the physical awareness derived from unsupervised learning. Additionally, after analyzing the nature of the garment self-collision problem, we introduce vertex repulsion-based constraints that effectively prevent conspicuous intersections within the mesh while preserving fine details. Finally, to ensure effective propagation of node information across the mesh, we propose a multi-scale graph processing technique for the garment deformation task, featuring structure-preserving pooling and unpooling strategies that significantly enhance result quality. The experimental results demonstrate that our method outperforms the state-of-the-art under multiple metric evaluations.

*Keywords:* Clothing deformation, Hybrid learning, Graph pooling, Attribution analysis

---

## 1. Introduction

The development of digital garments with realistic behaviors is a critical area in computer graphics, serving industries such as film, gaming, and fashion design. Traditional methods like skinning-based deformation [1, 2] and physics-based simulation [3, 4, 5] provide effective cloth animation solutions but often require a compromise between realism and computational efficiency. Learning-based approaches [6, 7, 8, 9] have shown promise in handling deformation tasks. However, their application is largely restricted to garments with fixed topologies, limiting the generalization capabilities. In response, recent advancements have explored the application of graph neural networks (GNNs) [10, 11, 12] to model clothing behavior. GNNs are particularly valued for their ability to generalize across varying mesh structures, enabling topology-agnostic adjustments. However, achieving physical realism continues to present challenges.

Currently, methods that can effectively incorporate diverse physical parameters (*e.g.*, gravity, repulsion, shear, and stretch forces) into a single framework for simulating various types of clothing are not yet available. This is primarily due to the complexities involved in managing the interactions between diverse forces, which pose significant challenges in training a neural

network capable of accurately modeling multiple physical laws. Our empirical findings suggest that training physics-informed neural networks with varied physics-based losses frequently leads to issues such as gradient vanishing and uncontrollable fluctuations in losses. Additionally, these physical constraints are typically applied in an unsupervised manner, which means the learning process lacks explicit, directive guidance. This absence of effective supervision further complicates the optimization process, making it difficult for current solutions to meet the multi-objective demands of the training. Consequently, models frequently struggle with parameter tuning, which in turn affects their ability to adapt to different garment types and weakens their generalization capabilities [13, 14, 15].

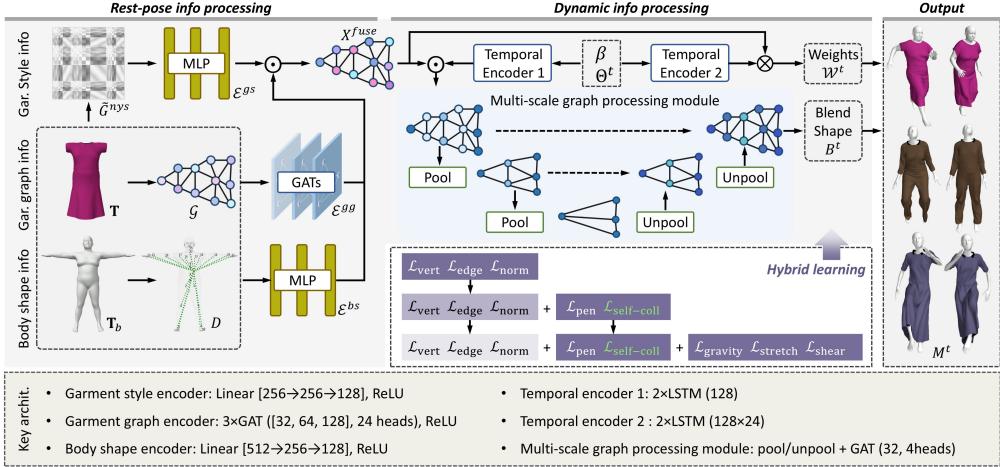
This phenomenon motivates us to explore a hybrid learning strategy that enables models to progressively master different physical laws. Instead of relying solely on direct unsupervised multi-objective optimization, we begin with supervised learning using the available simulation data. This initial phase establishes a plausible baseline by applying constraints on vertices and normal vectors, capturing key relationships in the data. Subsequently, we transition to integrating various physics-based constraints during the unsupervised learning phases. Here, we first apply different collision losses to equip the model with capabilities for detecting and preventing collisions. Following this, we introduce distinct mechanics-related constraints, further enhancing the model’s ability to effectively perceive and adapt to physical laws.

Within this hybrid learning framework, we design a novel self-collision constraint that primarily mitigates collisions by adjusting the spacing between vertices that are proximate in the occurring fabric fold. While this approach does not completely eliminate the self-collision issue, it has proven effective in removing the conspicuous ones that are easily noticeable to human eyes. Currently, achieving complete self-collision prevention remains computationally intensive, and, to the best of our knowledge, end-to-end self-collision elimination is not yet feasible with deep learning tools alone.

Moreover, in terms of neural network architecture, to improve the information propagation and exchange capabilities of GNNs, we incorporate a multi-scale module within the graph processing workflow. This module, which includes both garment graph pooling and unpooling processes, facilitates enhanced interactions between garment mesh vertices. As a result, it can more accurately generate dynamic garment skinning weights and blend shapes within the standard skinning pipeline. The overview of our pipeline is depicted in Fig. 1.

Specifically, we initially apply different encoders to handle multi-source information in the rest pose and then combine them to generate the fused graph representation. In the dynamic information processing stage, our pipeline contains two branches: for skinning weights, a temporal encoder processes the motion sequence to extract state features, which are then combined with the fused graph features to compute the skinning weights; for blend shapes, another temporal encoder deals with motion and body shape, and these features, together with the fused graph representation, are further processed by the multi-scale graph module to generate the blend shapes. Finally, the skinning function utilizes the computed skinning weights and blend shapes to generate the final deformation. In conclude, our contributions are as follows:

- We introduce a hybrid learning approach aimed at generating controllable and physics-aware garment deformations. This strategy effectively leverages the strengths of both supervised and unsupervised learning, while mitigating their respective limitations.
- We design a self-collision constraint to prevent conspicuous intersections by analyzing critical collision scenarios. This method effectively eliminates artifacts without causing detail loss or over-smoothing, ensuring realistic outcomes.



**Fig. 1.** The overview of our deformation network. The network operates as an end-to-end system, taking input in the form of garment, body, and motion sequences, and undergoes rest-pose information processing and dynamic information processing to generate deformation results.

- We introduce a multi-scale graph processing module for garment deformation, which preserves mesh connectivity during coarsening and utilizes original node information in refinement. This method enhances information diffusion and improves both learning performance and deformation quality.

A preliminary version of our work was previously published in a conference proceeding [16]. This journal paper builds upon that foundation with two additional contributions. First, we introduce and elaborate on our novel hybrid learning scheme, complemented by experiments that demonstrate its stability and effectiveness in learning garment deformation. Second, we tackle the self-collision limitation stated in the earlier version by offering a comprehensive analysis of the collision problem, presenting a new formulation, and providing experimental validation to support its efficacy.

## 2. Related Work

### 2.1. Physics-based Simulations

Physics-based simulations (PBS) are widely recognized for their ability to achieve a high degree of realism in deformation effects [17, 18, 19]. They leverage the discretizations of classical mechanics to accurately simulate the deformation process of cloth. Despite their impressive capability to generate realistic deformations, they inherently come with high computational demands, require extensive computational resources, and pose significant challenges in reaching interactive speeds [20, 21]. To enhance efficiency, strategies such as leveraging modern GPU parallel computational capabilities [22, 23, 24], focusing on constrained scenarios [25], and incorporating high-frequency wrinkles into low-frequency meshes [26, 27] have been employed. Nevertheless, even with these advancements, these approaches remain inadequate when faced with limited computational capacity. Despite research attempts to introduce perceptual control [28] and the automatic inference of physical parameters [29, 30, 31], these methods are restricted

to function optimally only in controlled environments. Consequently, PBS methods continue to demand substantial time for upfront parameter tuning and demand a high level of expertise in parameter adjustment.

## 2.2. Learning-based Methods

Learning-based methods in clothing animation, which predict garment deformations from input parameters in 2D [32, 33] or 3D space [8, 34], have gained popularity due to their efficiency and automation compared to traditional physics-based simulations. Building on pose space deformation [2], research has explored learning deformations influenced by factors like pose, shape, and garment size [35, 36, 37, 7], although challenges remain in accurately predicting folds for loose garments. To improve detail in garment appearance, some studies have redefined physics-based simulations into optimization problems with physics-derived loss terms and introduced unsupervised learning to avoid reliance on ground truth data [13, 15, 14, 38]. Recent advancements have also improved deformation predictions for loose-fitting dresses [8]. However, a shared limitation of all the aforementioned methods is their inability to generalize to garments with different mesh structures.

Recent studies [39, 40] addressing model generalization issues have turned to graph neural networks (GNNs) due to their adeptness in handling 3D data without prior mesh graph structure knowledge [41]. DANet [12] develops a unified framework based on the graph attention network (GAT) [42] to model nonlinear garment deformations with fine details for different SMPL [43] bodies. Other approaches include GraphUNet-based [10, 11] and PointNet-based frameworks [44] for cloth deformation, yet they still face limitations in topology variation and in effectively using global information to generalize well to new garments or poses. Techniques from computer vision focus on clothing deformation, incorporating processes like fold transfer [45], for virtual try-ons [46, 47]. While these methods are capable of producing photo-realistic outcomes, they frequently encounter difficulties in accurately capturing 3D shape details and are sensitive to environmental variables.

Recent advances in optimization and learning further inform this field. Lagrange elementary optimization [48] and rigorous statistical evaluation methods [49] highlight the importance of adaptive algorithms and robust statistical testing in complex, multi-objective scenarios. Additionally, studies on machine learning for structured and heterogeneous data, such as demographic-aware insomnia prediction [50] and self-growth learning-based scheduling in manufacturing [51], demonstrate the effectiveness of combining deep models with domain-specific constraints for improved generalization and real-world applicability. These insights motivate our integration of hybrid learning and statistical optimization within graph-based garment modeling.

## 2.3. Graph Pooling Methods

Graph Pooling Methods, inspired by pooling layers in convolutional neural networks, are designed to generate coarser sub-graphs that enhance message propagation through hierarchical representation. These characteristics are crucial for simulating cloth force propagation and have directly informed the design of our method. Notable recent developments in this field include DiffPool [52], TopKPool [53], SAGPool [54], ASAP [55], and rasterization-based pooling methods [56]. Although these methods have introduced notable innovations, they still face common challenges, including inefficient pooling and difficulties in preserving stable mesh structures, both of which are crucial for maintaining the integrity of 3D models and other graph-based analyses.

### 3. Model Architecture

#### 3.1. Garment Deformation Model

We initiate with a garment mesh template  $\mathbf{T} \in \mathbb{R}^{N \times 3}$  consisting of  $N$  vertices, a SMPL basis body  $\mathbf{T}_b \in \mathbb{R}^{N_b \times 3}$  with  $N_b$  vertices and shape parameter  $\beta$ , and a sequence of poses  $\Theta^t = \{\theta^t, \dots, \theta^{t-m+1}\}$ , our goal is to learn a network that maps these variables to garment temporal skinning weights  $\mathcal{W}^t \in \mathbb{R}^{N \times S}$  ( $S$  is the number of joints in the body skeleton) and blend shapes  $B^t \in \mathbb{R}^{N \times 3}$ :  $\{\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta\} \rightarrow \{\mathcal{W}^t, B^t\}$ . Next, by applying a skinning function  $W(\cdot)$ , the garment deformation at  $t$ -th frame  $M^t$  is:

$$M^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta) = W(T^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta), \theta^t, \mathcal{W}^t), \quad (1)$$

$$T^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta) = \mathbf{T} + B^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta), \quad (2)$$

where  $T^t$  is the deformed mesh in the rest pose, obtained by adding the predicted blend shape  $B^t$  to  $\mathbf{T}$ . We next describe how to achieve the above objective in detail.

#### 3.2. Deformation Approximation

##### 3.2.1. Multi-source information processing

Garments come in a wide variety of shapes and topologies, with flexible vertex-to-vertex interactions during deformation. We model this complexity using a graph  $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ , where  $\mathcal{X}$  contains features for  $N$  vertices detailing position, normal, and joint distances, and  $\mathcal{E}$  represents the edges. Then, we employ a garment graph encoder  $\mathcal{E}^{gg}$ , which transforms the node features  $X = [x_1, \dots, x_N]^\top$  into higher-level latent representations using several graph attention blocks, resulting in the refined encoded graph  $\hat{\mathcal{G}}^{gg}$  with features  $X^{gg}$  (see the left center of Fig. 1).

To capture the global characteristics of a garment style, we first construct an affinity matrix  $G \in \mathbb{R}^{N \times N}$ , where each element  $G_{ij}$  encodes the geodesic distance-based affinity between vertices. However, directly operating on  $G$  is computationally intensive and impractical for meshes with varying vertex counts and topologies. To overcome these challenges, we employ the Nyström algorithm [57] to obtain a low-rank approximation  $\tilde{G}^{nys} \in \mathbb{R}^{N^* \times N^*}$  (see the upper left corner of Fig. 1), where  $N^* \ll N$  and  $N^*$  is fixed for all garments. This approach offers two key advantages. First, it greatly improves computational efficiency while preserving essential style information. Second, it produces a unified, fixed-size representation that is invariant to mesh topology. As a result, this method eliminates the bias caused by mesh complexity and enables consistent modeling and comparison of diverse garment styles. We extract the leading eigenvalues of  $\tilde{G}^{nys}$  as global shape descriptors, which are then passed to the style encoder  $\mathcal{E}^{gs}$ , a multi-layer perceptron (MLP) that transforms them into a fixed-length style cue vector  $C^{gs}$ . These encoded style cues are subsequently integrated with  $X^{gg}$  and further refined by a self-attention block to produce the enhanced features  $\hat{X}^{gg}$ .

Body shape information also impacts garment deformation, with larger bodies exhibiting tighter fits and denser folds, and slimmer bodies showing more dynamic and expansive folds. To quantify these effects, we measure joint-vertex distances, represented by feature  $D \in \mathbb{R}^{N_b \times S}$  for body template  $\mathbf{T}_b$  (see the lower left corner of Fig. 1). These features are then processed by a body shape encoder  $\mathcal{E}^{bs}$ , using a MLP to transform them into body-related latent cues  $C^{bs}$ .

To synthesize comprehensive garment deformation-related information, we combine cues from both body and garment into a fused graph  $\mathcal{G}^{fuse}$ , forming the features  $X^{fuse} = \hat{X}^{gg} \odot C^{bs}$ , where  $\odot$  is element-wise multiplication.

### 3.2.2. Skinning weights and blend shapes generation.

For skinning weight generation, we process a motion sequence  $\Theta^t = \{\theta^t, \dots, \theta^{t-m+1}\}$ , where each  $\theta$  details the axis-angle and translation from the previous frame, using a temporal encoder  $\mathcal{E}^{t1}$  built with LSTM to extract state features. These are then merged with graph features to compute the skinning weights:

$$\mathcal{W}^t = X^{fuse} \otimes \mathcal{E}^{t1}(\Theta^t), \quad (3)$$

where  $\otimes$  denotes the multiplication.

Generating blend shapes, which account for detail wrinkles, is more complex than computing skinning weights. We process body shape  $\beta$  and motion  $\Theta^t$  to determine the velocity states  $\mathbf{V}^t$  of body vertices. These velocities are processed by another temporal encoder  $\mathcal{E}^{t2}$ . The output, combined with flattened  $X^{fuse}$ , forms integrated motion-related graph features  $\tilde{X}^{fuse}$ . This graph  $\tilde{\mathcal{G}}^{fuse}$  is input into a multi-scale processing module to generate the blend shapes:

$$B^t = \mathcal{M}^{mp}(\tilde{\mathcal{G}}^{fuse}). \quad (4)$$

Here,  $\mathcal{M}^{mp}$  employs a multi-scale structure to capture detailed deformations, with further specifics provided in subsequent sections.

### 3.3. Multi-Scale Graph Processing Module

Garment deformation modeling needs to effectively capture the propagation of information within the mesh and its impact on detailed deformations. Therefore, a deformation model should ensure that the influence of a vertex spreads to as many neighboring vertices as possible during feature propagation to prevent the ‘locality problems.’

Graph pooling plays a key role in facilitating the rapid diffusion of vertex information. However, unlike in typical graph learning tasks, garment graph pooling should preserve the structural connectivity of the entire garment. This preservation ensures consistency and stability in the mesh structure, enabling accurate and realistic simulations under various physical conditions. The corresponding unpooling process should carefully reintegrate surrounding vertex information to accurately restore the graph’s fine details. Additionally, as vertices might come into close physical contact due to bending and shearing during deformation, it is critical that the graph pooling approach minimizes reliance on Euclidean spatial proximity. Based on these considerations, we use the garment graph information itself to design the pooling and unpooling strategies.

Our approach is inspired by the preservation of second-order connections in a directed acyclic graph. The detailed pooling and unpooling algorithms are provided in the supplementary material. As depicted in the pooling part of Fig. 1, we alternate depths to remove vertices that are current first-order neighbors. To partition a garment mesh, we first identify a seed vertex  $e^{ecc}$  based on the eccentricity of the garment graph  $\mathcal{G}^0$  in its rest pose. We measure the geodesic distances from this seed to all other vertices and then selectively remove those with odd geodesic distances to form the pooled vertex set  $X^c$ . It is important to clarify that, in our context, the term ‘geodesic distance’ is used in the sense of graph theory, referring to the shortest path between two vertices as measured by the number of edges along the minimal connecting path on the mesh graph, rather than a surface metric distance. To maintain original connectivity, edges are added between vertices in  $X^c$  that were indirectly connected via the removed vertices. Edges are further added to the retained coarse vertex set based on the ‘from’ and ‘to’ vertices connected to the removed vertices, where the vertices with connection relationships in the original graph are also connected.

For unpooling, we enhance vertex feature restoration by using weighted filling based on the neighbor information  $A$  of the finer graph, which helps eliminate wave-like artifacts. In our notation, we denote the existence of an edge  $(i, j)$  with  $A_{ij} = 1$ . Our process starts by identifying neighbor relationships  $\mathcal{N}^{r0}$  from the removed vertices and  $\mathcal{N}^{c0}$  from the coarse graph vertices. We then compute the weight  $\hat{w}_{ij}$  and directly integrate coarse vertex features into the fine graph. Following this, we normalize  $\hat{w}_{ij}$  and apply the edge weight  $w_{ij}$  to weight the features of neighboring vertices, reconstructing those of the removed vertices.

Building on our proposed pooling and unpooling strategies for garment meshes, we design a multi-scale processing module  $\mathcal{M}^{mp}$  for generating detailed blend shapes  $B'$ . By integrating these blend shapes with approximated dynamic skinning weights, we efficiently achieve the final garment deformation  $M'$  (as shown in the right part of Fig. 1).

#### 4. Hybrid Learning Framework

We propose a hybrid progressive learning scheme for our garment deformation model, combining the advantages of both supervised and unsupervised learning methods. Supervised learning provides precise guidance for model optimization but often relies heavily on extensive data, which can lead to overfitting and reduced efficacy on novel objects. On the other hand, unsupervised learning methods, which employ physical or geometric constraints, lessen the dependency on pre-existing data but struggle with the complex management of multiple loss terms, potentially leading to uncontrolled output.

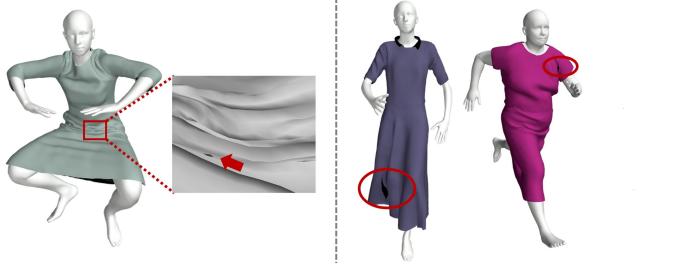
Our hybrid learning scheme combines the strengths of supervised and unsupervised learning across three progressive phases. Initially, supervised learning establishes a solid foundation, simplifying early parameter tuning and setting a robust baseline for the model (phase 1). We then transition into unsupervised learning, implementing collision-related constraints that guide the model towards plausible clothing behavior and introducing a novel self-collision loss function to mitigate conspicuous collision artifacts (phase 2). Finally, we incorporate mechanics-related loss terms to further enhance the model's physics awareness, ensuring that it both appears realistic and adheres to physical principles (phase 3). Next, we will provide a detailed description of each phase.

##### 4.1. Learning Phase 1

In our hybrid learning framework, we start with supervised learning, employing vertex positions  $\mathcal{L}_{\text{vert}}$ , edge lengths  $\mathcal{L}_{\text{edge}}$ , and normal angles  $\mathcal{L}_{\text{norm}}$  as constraints, which have been typical choices in supervised training. This enables the model to steadily converge to a stable state, accurately producing deformations within the training data distribution. Specifically, We utilize L2 loss for both vertex and edge attributes to minimize the Euclidean distances and squared differences between predictions and ground truth, ensuring spatial accuracy and structural integrity. Additionally, cosine similarity loss is applied to normal angles, promoting correct geometric orientations. For these components, scaling factors are set as 100, 15, and 50, respectively.

##### 4.2. Learning Phase 2

To mitigate overfitting, we progress to the second phase of our training scheme once the distance error stabilizes. At this point, we introduce unsupervised components by activating penetration and self-collision losses and freeze the rest-pose processing module. Concurrently, we reduce the weight on supervised loss terms by a scaling factor of 0.05. The penetration loss is



**Fig. 2.** Inconspicuous (left) and conspicuous (right) self-collisions.

derived from (**author?**) [44] that has gained widespread adoption in many garment deformation studies [58, 13, 11, 14]. The penetration loss is defined as:

$$\mathcal{L}_{\text{pen}} = k_c \sum_i^N \min(l(v_i, f_j) - \epsilon, 0)^2, \quad (5)$$

where  $l(v_i, f_j)$  is the distance between garment node  $i$  with the closest face of the body mesh and  $\epsilon$  is a penetration threshold value.  $k_c$  is set at 1.5.

In addressing self-collisions within garment deformations, we initially classify them into inconspicuous and conspicuous types based on experimental observations (Fig. 2). Due to the densely clustered wrinkles in certain areas, self-collisions are inevitable in neural simulated garment deformations. Removing all such collisions could lead to overly smoothed details, which is undesirable for maintaining realism. In dynamic scenarios, these inconspicuous collisions are typically less noticeable and, therefore, can be pragmatically ignored. However, conspicuous self-collisions, such as those occurring between different garment parts like front and back hemline intersections, are visually significant and can detract from the aesthetic and functional qualities of the garment. Our self-collision loss function is specifically designed to mitigate these types of collisions, which is defined as:

$$\mathcal{L}_{\text{self-coll}} = \frac{k_{sc}}{\sum_{i,j} C_{i,j}} \sum_{i,j} C_{i,j} e^{-\lambda_a(\|v_i - v_j\|_2 - \lambda_b)}, \quad (6)$$

where  $k_{sc} = 2$ .  $\lambda_a$  and  $\lambda_b$  representing the attenuation velocity and offset, respectively, are adjusted based on the fabric thickness. For instance, with a fabric thickness of 2 mm, we set  $\lambda_a = 8$  and  $\lambda_b = 0.4$ . Fundamentally, this loss introduces repulsive forces between vertices using the connection matrix  $C$ .  $C_{ij} = 1$  if both conditions are met: a) The geodesic distance between vertices  $i$  and  $j$  in the rest pose exceeds a threshold. b) The Euclidean distance between vertices  $i$  and  $j$  is less than another threshold. These conditions ensure that the repulsive force is applied only when vertices, initially geodesically distant, become undesirably close in Euclidean distance during deformation. The thresholds are adjustable based on fabric properties; here, they are set to 10cm and 2cm for geodesic and Euclidean distances, respectively. If either condition is not met,  $C_{ij} = 0$ , and no repulsive force is applied. This loss function effectively alleviates self-collisions, such as those occurring in areas like sleeve-body or leg-leg intersections.

#### 4.3. Learning Phase 3

When the collision loss stabilizes, we transition to the third phase of our training scheme by activating cloth model and gravity losses. Concurrently, we adjust the hyperparameters of

supervised loss terms by reducing their influence with a scaling factor of 0.2. Various cloth model losses can be employed to achieve realistic simulations; a common choice is the Saint Venant Kirchhoff (StVK) elastic material model [14]. In our method, we opt for [59, 15], as it seemingly facilitates convergence by directly manipulating the shear stiffness during training. The losses that are finally enabled include:

$$\mathcal{L}_{\text{gravity}} = -k_g \sum_i^N m_i v_i \cdot g, \quad (7)$$

$$\mathcal{L}_{\text{stretch}} = k_s \sum_{i=1}^{N_F} a_i \left( L_\delta(\|w_i^u\|_2 - b^u) + L_\delta(\|w_i^v\|_2 - b^v) \right), \quad (8)$$

$$\mathcal{L}_{\text{shear}} = k_h a w^u \cdot w^v, \quad (9)$$

$$\begin{pmatrix} w^u & w^v \end{pmatrix} = \begin{pmatrix} \Delta x_1 & \Delta x_2 \end{pmatrix} \begin{pmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{pmatrix}^{-1}, \quad (10)$$

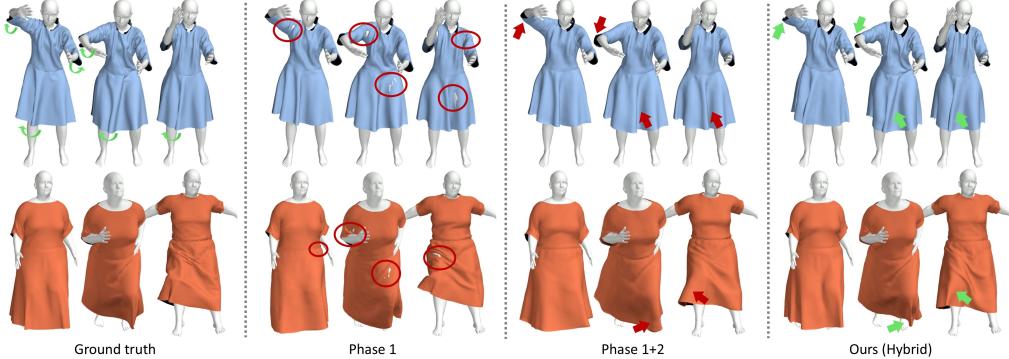
where  $m_i$  is the vertex mass and  $g = [0, 0, -9.8]$  is gravitational acceleration.  $a_i$  is the  $i$ -th triangle area in  $uv$  coordinates and  $N_F$  is the face number.  $L_\delta$  represents the Huber loss function with its parameter  $\delta$  set to 0.05. This parameterization is utilized to manage error sensitivity and robustness within the simulation.  $b^u$  and  $b^v$  are set to 1 can be used to modify stiffness.  $w^u$  and  $w^v$  are the deformation gradients of in the  $u$  and  $v$  directions. The vertex position differences are  $\Delta v_1 = v_j - v_i$  and  $\Delta v_2 = v_k - v_i$ , where indices  $i, j, k$  are face triangle vertices. The term  $\|w_i^u\|_2$  and  $\|w_i^v\|_2$  measure the magnitudes of these gradients. The vertex position  $v_i$  changes in world space, and the fixed plane coordinate is represented as  $(u_i, v_i)$ .  $\Delta u_1, \Delta u_2, \Delta v_1, \Delta v_2$  are similar to  $\Delta x$ . The hyperparameters  $k_g, k_s$ , and  $k_h$  are set to 0.1, 5, and 2, respectively.

Note that the learning in phase 3 still incorporates elements from phase 2 (see the bottom of Fig. 1). If we reverse their order and allow phase 3 to advance, the model might attempt to apply physical laws to unstable or incorrect collision states, thereby compounding errors rather than correcting them.

## 5. Experiments

### 5.1. Implementation details

For the dataset, we collect various types of garments from CLOTH3D [60], human body from SMPL [43], and continuous motions from CMU Mocap dataset [61]. We then simulate garment-body pairs using silk-like fabrics in Blender. The training set consists of 55 garments and nine body shapes, totaling approximately 50,000 poses. The validation set contains five garments and three body shapes, with around 3,000 poses. The test set contains 15 garments with randomly generated body shapes, with around 8,000 poses. There is no overlap among these datasets, ensuring their independence. To mitigate potential dataset imbalance and garment-style bias, both the training and test sets include dresses and jumpsuits with diverse style variations, such as different sleeve lengths, dress lengths, trouser lengths, and garment fit (ranging from loose to tight). Body shapes are sampled to cover a wide range of heights, weights, and gender characteristics. Each garment maintains distinctive features while sharing structural similarities, which facilitates effective learning and generalization of garment deformation. In addition, the test set is designed to include some out-of-distribution samples, such as garment styles or attribute combinations that do not appear in the training set. This allows us to more rigorously assess the



**Fig. 3.** Comparison of deformations across different learning phases. Our hybrid learning model significantly enhances the realism and detail of dynamic garment deformations.

generalization and robustness of our method. For further details on the network architecture and implementation, please refer to the supplemental materials.

### 5.2. Quantitative evaluation metrics

We use the following five quantitative quality measures:

$$E_{\text{verts}} = \frac{1}{N} \sum_{i=1}^N \|v_i - v_i^{\text{gt}}\|, \quad (11)$$

$$E_{\text{vnorm}} = \frac{1}{N} \sum_{i=1}^N \arccos \left( \frac{n_i \cdot n_i^{\text{gt}}}{\|n_i\| \|n_i^{\text{gt}}\|} \right), \quad (12)$$

$$E_{\text{fnorm}} = \frac{1}{N_F} \sum_{j=1}^{N_F} \arccos \left( \frac{f_j \cdot f_j^{\text{gt}}}{\|f_j\| \|f_j^{\text{gt}}\|} \right), \quad (13)$$

$$E_{\text{edge}} = \frac{1}{N_E} \sum_{k=1}^{N_E} \frac{|l_k - l_k^{\text{gt}}|}{l_k^{\text{gt}}}, \quad (14)$$

$$E_{\text{curv}} = \frac{1}{N} \sum_{i=1}^N |\kappa_i - \kappa_i^{\text{gt}}|. \quad (15)$$

Here,  $N$ ,  $N_F$ , and  $N_E$  denote the numbers of vertices, faces, and edges, respectively.  $v_i$  represents the position of the  $i$ -th vertex;  $n_i$  and  $f_j$  are the normals of the  $i$ -th vertex and the  $j$ -th face, respectively;  $l_k$  is the length of the  $k$ -th edge; and  $\kappa_i$  denotes the discrete Gaussian curvature at vertex  $i$ . The superscripts “gt” refer to the ground-truth values.

### 5.3. Evaluation on Hybrid Learning

To demonstrate the advantages of our proposed hybrid learning scheme, we conduct an experiment where we retained only the initial and the first two learning phases.

In Fig. 3, when all unsupervised losses are eliminated, the garment achieves a general approximation of its basic deformation. However, this configuration leads to unrealistic behaviors,

**Table 1:** Quantitative deformation results of different phases. We measure five metrics: the average vertex distance  $E_{\text{verts}}$  (mm), the average angular deviation of vertex normals  $E_{\text{vnorm}}$  ( $^{\circ}$ ), that of face normal  $E_{\text{fnorm}}$  ( $^{\circ}$ ), the relative edge length error  $E_{\text{edge}}$  (%), and discrete Gaussian curvature error  $E_{\text{curv}}$  between predictions and PBS.

Methods	$E_{\text{verts}}$	$E_{\text{vnorm}}$	$E_{\text{fnorm}}$	$E_{\text{edge}}$	$E_{\text{curv}}$
Phase 1	45.23	28.95	32.81	15.24	0.062
Phase 1+2	33.36	21.40	24.15	11.33	0.046
Ours (hybrid)	<b>24.36</b>	<b>16.81</b>	<b>19.10</b>	<b>9.89</b>	<b>0.035</b>

including obvious collisions between the body and the garment, and exhibits unnatural rigidity in movements. Next, after progressing through two phases of learning, the model becomes more effective at addressing collisions, resulting in reduced penetration and intersection within the mesh. However, the dynamic qualities remain underdeveloped. Observe the green arrows in the ground truth indicating that loose elements such as cuffs and hemlines should exhibit some degree of swing. Yet, the results after phase 2 still appear relatively static. In contrast, when implementing our proposed hybrid training scheme, which includes three progressive phases, the predicted garments closely resemble the ground truth and exhibit enhanced controllability and physical realism. The quantitative results presented in Table 1 further support this, demonstrating that our hybrid learning scheme achieves the lowest error rates.

We also experimented with maintaining only the unsupervised components or altering the sequence of the learning phases. These changes, however, compromised the model’s ability to train stably or converge effectively, resulting in unreliable outcomes. The above experiment demonstrates the validity of our designed progressive strategy. Initially, the model is guided to closely mimic the ground truth through supervised learning. Then, collision-relevant losses are introduced to enhance the plausibility. Finally, unsupervised losses that align with the physical laws governing fabric dynamics are incorporated, improving the realism of the clothing simulations.

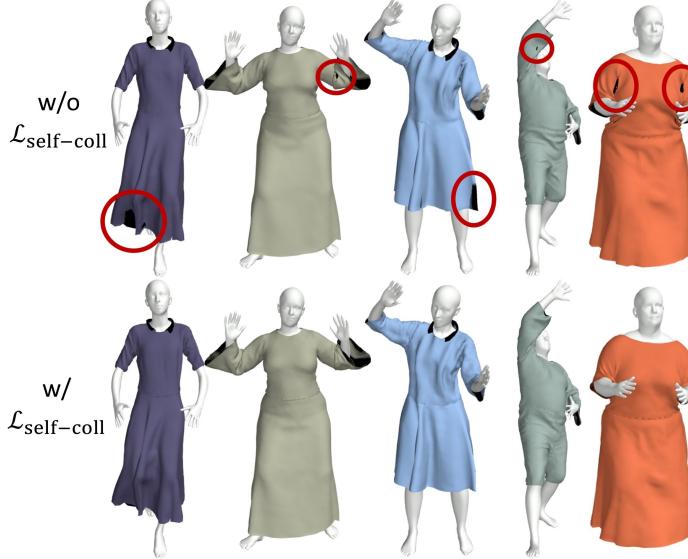
#### 5.4. Evaluation on Self-Collision Constraint

In this section, we evaluate the effectiveness of our proposed self-collision constraint  $\mathcal{L}_{\text{self-coll}}$  across various unseen data. Our evaluation focuses on comparing the performance of garment deformations with and without the implementation of the self-collision loss term.

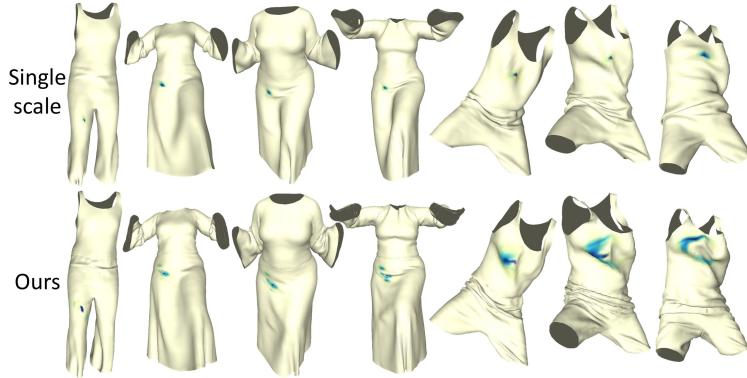
As demonstrated in Fig. 4, our observations reveal that, in the absence of this constraint, specific areas of garments such as the front and back hems of the dress, the underarm area alongside the upper-arm sleeves, and the elbow regions connecting the upper-arm to forearm sleeves, are particularly susceptible to collisions during dynamic movements. These collision events frequently lead to noticeable artifacts. By incorporating the self-collision loss into the model, we enable a more refined interaction among mesh vertices, allowing the simulation to more closely mimic the physical laws that govern real-world fabric behavior. The self-collision loss specifically introduces repulsive forces between pairs of vertices that are potentially colliding, effectively preventing vertices from coming too close or penetrating each other. This mechanism is particularly advantageous for simulating complex movements and poses, ensuring that different parts of the garment maintain appropriate separation throughout motion.

#### 5.5. Evaluation on Multi-Scale Architecture

The standard graph-attention-based model aggregates vertex features from immediate neighbors, with the extent of influence a single vertex can exert being limited to the network struc-



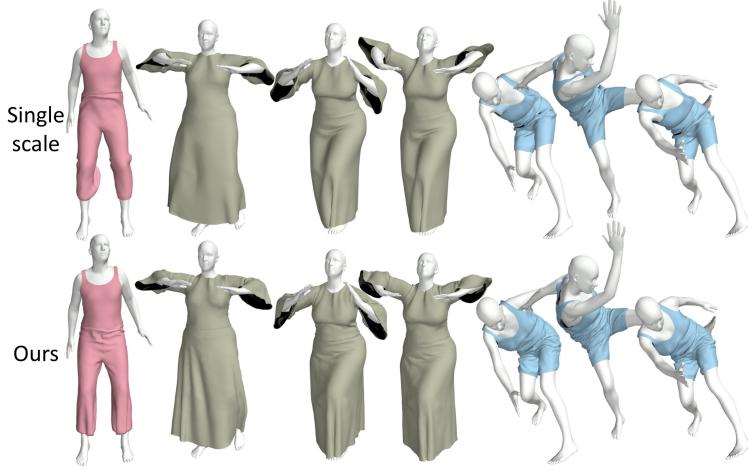
**Fig. 4.** Comparison of deformations without (w/o) and with (w/) unsupervised self-collision constraints.



**Fig. 5.** The attribution results of the single-scale network and ours. The deformation result in our network always receives a broader impact from ambient vertices.

ture's depth. This often leads to insufficient interactions between vertices, thus hindering realistic clothing simulations. To overcome this, we introduce a multi-scale processing module that incorporates graph pooling and unpooling strategies.

While pooling might intuitively be perceived as advantageous for the neural simulation of garment deformation, we cannot analyze the influence field of a vertex in practice. Thus, we incorporate into the experiment the method of attribution calculation, a technique commonly used within the realm of neural network interpretability research. At its core, the attribution method computes the contribution of vertices in the input graph that bear an impact on a particular output result. If some vertices have a substantial impact on the deformation results, these vertices will be highlighted in the heatmap generated by the attribution value calculations.



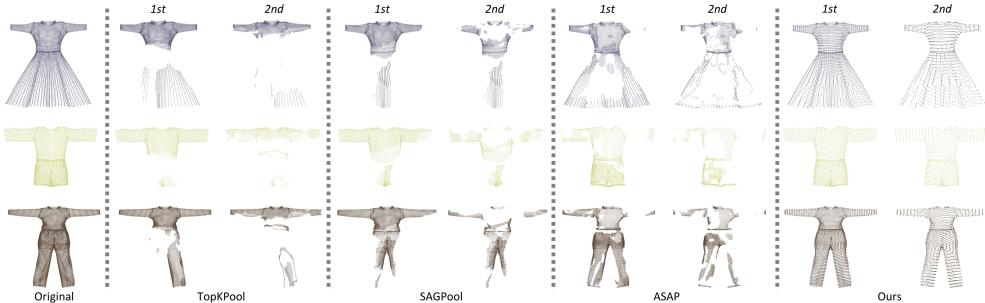
**Fig. 6.** The deformation results of the single-scale network and our multi-scale method.

More specifically, we use an attribution calculation method based on Aumann-Shapley values [62]. This method incrementally introduces input features, calculating and accumulating gradients at various stages, thereby deriving the causal correlation between input and output. For the sake of simplified representation, we primarily focus on the graph feature  $X$  within the model input. The output  $f_i(\cdot)$  represents the position of the vertex  $i$  in the conclusive deformation prediction, where  $f$  stands for the deformation network. Then, we have:

$$R_i = X \sum_{k=1}^{N_s} \frac{1}{(1 - \xi_k^2) [P'_{N_s}(\xi_k)]^2} \frac{\partial f_i(1/2(1 + \xi_k)X)}{\partial X}, \quad (16)$$

where  $N_s$  is the number of sample points in Gauss-Legendre quadrature for approximating the definite integral.  $\xi_k$  is the quadrature point of the  $k$ -th Legendre polynomial.  $P'_{N_s}$  is the derivative of Legendre polynomials at the sample point. In our experiments,  $N_s$  is set to 50, and  $R_i$  is the attribution result which has the same size with the input  $X$ . The heatmap shown in Fig. 5 is generated using attribution result dimensions corresponding to normals.

To thoroughly evaluate the effects of multi-scale architecture, we construct a network devoid of pooling and unpooling for comparison. In this setup, the graph attention blocks involved in multi-scale processing module are cascade connected to produce a single-scale network. The attribution results in Fig. 5 show the factual field influencing the vertex, assisting in the examination of specific deformation regions. As depicted, the vertices in our network receive a broader impact, resulting in a more comprehensive deformation. In the single-scale structure, the vertices contributing to deformation primarily cluster around the vertex under analysis. This concentration can invariably impact the simulation performance of forces involved in deformation. As shown in Fig. 6, the single-scale deformation of the green dress does not follow the motion to create a coordinated wrinkle effects, such as the shoulders and hemline areas. Conversely, our deformation results tend to demonstrate superior consistency overall and can generate reasonable details on tested garments. Furthermore, we observe that increasing the number of pooling layers beyond two does not provide additional benefits for garment deformation quality. Instead, it leads to unnecessary loss of fine-grained information and increases computational time.



**Fig. 7.** Structure preservation between different graph pooling strategies. Starting from the original template dress, we show the results for pooling once and twice. This figure is best viewed on screen.

### 5.6. Evaluation on Different Pooling Strategies

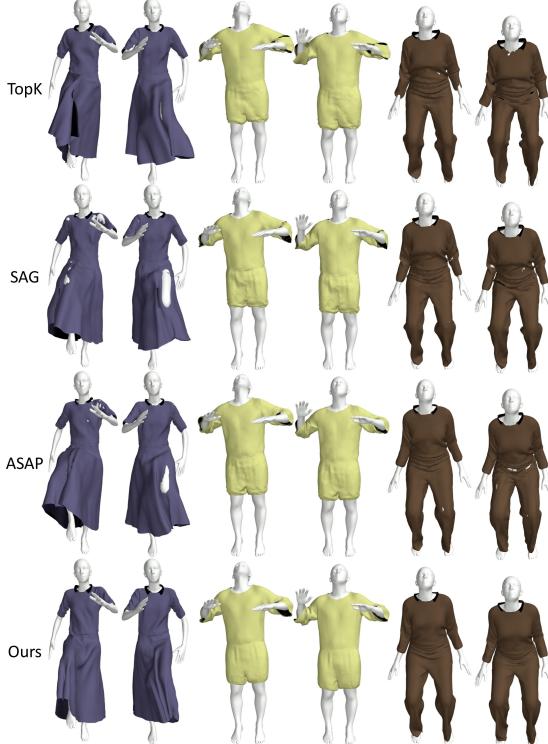
In this section, we conduct an analysis of various mesh pooling strategies for garment deformation tasks, comparing our structure-preserving pooling method with TopKPool [53], SAGPool [54], and ASAP [55].

As illustrated in Fig. 7, our approach pools nodes at every alternate level of a breadth-first-search traversal, thus achieving mesh sparsification while maintaining the global structure after one and two rounds of pooling. This technique offers a straightforward and effective way of representing the garment mesh comprehensively. Importantly, it ensures that no specific area of the mesh is disproportionately ignored, facilitating a more even preservation of the mesh structure across the entire garment. The other three methods are learnable pooling strategies. They either retain the top-k nodes based on a scalar projection score, use a self-attention mechanism to select important nodes, or utilize a structure-aware score to choose which nodes to keep. These methods can inadvertently omit vertices from specific areas, potentially resulting in the loss of important structural regions. While ASAP effectively maintains mesh boundaries, it is less successful at preserving the interior structure. Overall, while these methods may be better suited for graph data such as protein or molecular graphs, where node importance significantly varies, our method excels in preserving mesh structure. This makes it particularly effective for tasks involving garment deformation.

Additionally, we present qualitative comparison of applying different pooling strategies in Fig. 8. Our pooling approach excels in capturing global features relevant to deformations across different garments, body shapes, and movements. This capability enables the model to generalize effectively to unseen test data and produce reliable deformation predictions. Alternatively, approaches that employ other learnable pooling methods necessitate the assignment of importance scores to nodes and subsequent pooling based on these scores. However, these scores are occasionally overfitted, which can hinder generalization and result in issues such as regional deformation artifacts and loss of detail in the final outputs. On the whole, our graph pooling strategy, specifically designed for garment simulation tasks, proves to be more effective in producing higher quality deformations.

### 5.7. Comparison to Existing Methods

In this section, we compare our approach against physics-based simulation (PBS) as well as three state-of-the-art methods: GAPS [38], DANet [12], and HOOD [11]. The primary distinctions among these methods are their abilities to generate realistic deformations on various



**Fig. 8.** Qualitative deformation results of applying different pooling strategies.

**Table 2:** Comparison with state-of-the-art methods in deformation characteristics, model generalization, and inference speed.

Methods	Dynamics	Generalization	Speed
GAPS	✓	✗	370.4
DANet	✗	✓	297.3
HOOD	✓	✓	13.4
Ours	✓	✓	402.1

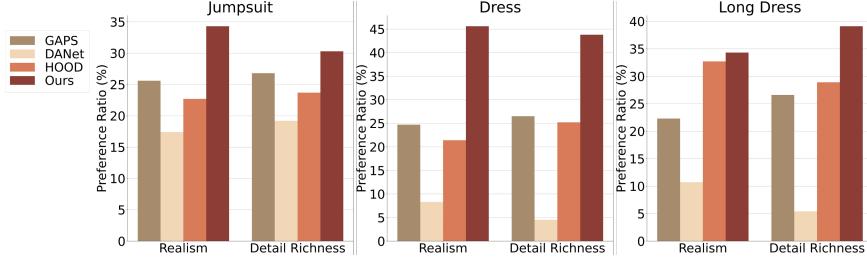
garments and their inference time, as depicted in Table 2. The inference speed evaluations were conducted on a computer equipped with an Intel i9-13900K CPU and an NVIDIA GeForce RTX 4090 GPU. GAPS shows robust generalization to novel motions but falls short in its ability to process diverse garments within a unified framework. DANet lacks dynamic processing capabilities, resulting in unnatural results when handling sequences of actions. In contrast, our method not only generates detailed and realistic deformations across a wide range of garments but also achieves this at hard real-time speeds. This enhances the practical utility of our approach, particularly in scenarios that require the simultaneous handling of multiple garment types with different topologies.

For inference time, we observed a substantial increase of more than 20% in runtime during



**Fig. 9.** Qualitative comparison with state-of-the-art approaches.

the replication of other methods. This discrepancy might be attributed to the backend architecture of the deep learning frameworks used. Consequently, we directly report the time performance results as presented in the original papers. GAPS, DANet, and our method all achieve performances that qualify as hard real-time. Both DANet and our method use graph learning for garment deformation; however, DANet relies on a two-step process for predicting deformations. In contrast, as demonstrated in Sec. 3.2, our model is compact and streamlined. It integrates specialized components for handling different types of features and uses temporal weights to directly compensate for deformation details, thereby enhancing the efficiency of deformation estimation. HOOD does not achieve similar efficiency, as it lacks the capability to process motion sequences in batches, diminishing the value of using deep neural networks. Another important difference is that GAPS and HOOD are completely unsupervised. Despite their advantage in reducing data



**Fig. 10.** User evaluations of the realism and detail richness in the garment deformations generated by the tested methods.

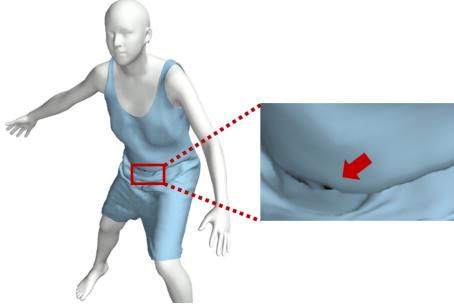
**Table 3:** Quantitative comparison with state-of-the-art approaches. Predictions are evaluated against PBS using five metrics on two types of garments: jumpsuits and dresses.

Methods	Jumpsuit					Dress				
	$E_{dist}$	$E_{vnorm}$	$E_{fnorm}$	$E_{edge}$	$E_{curv}$	$E_{dist}$	$E_{vnorm}$	$E_{fnorm}$	$E_{edge}$	$E_{curv}$
GAPS	34.53	21.24	25.39	11.76	0.048	56.54	36.97	41.19	19.72	0.078
DANet	39.15	25.57	28.74	13.53	0.056	65.29	42.84	47.66	21.40	0.089
HOOD	28.64	18.71	21.03	10.61	0.039	49.51	32.18	36.15	16.83	0.069
Ours	<b>20.29</b>	<b>13.55</b>	<b>15.85</b>	<b>9.06</b>	<b>0.028</b>	<b>28.43</b>	<b>20.06</b>	<b>22.34</b>	<b>10.71</b>	<b>0.041</b>

preparation compared to our method, their lack of effective evaluation indicators during training makes it challenging to control the overall training progress.

The experimental results in Fig. 9 show our method not only maintains dynamic effects but also achieves the realistic detail. Specifically, for the jumpsuit, our method produces more diverse wrinkle details on the sleeves and accurately demonstrates the influence of gravity. Although GAPS and HOOD also include gravity loss, their deformation effect on the sleeves remain less realistic. This highlights a common challenge in unsupervised learning: balancing gravity and inertia losses can make the optimization process unstable, especially when multiple physical forces are involved. For the pink dress, our method also produces the richest details and shows significant dynamics at the hem. The examples of the long dress are a series of continuous motions, providing a clear reference for comparing the dynamic effects of different methods. All methods, except for DANet, display obvious dynamics. DANet's lack of a dynamic processing function results in deformations that appear stiff. Overall, our method consistently produces high-quality deformation details across various garments and motions, and achieves results that are closer to the real PBS compared to other state-of-the-art methods. Furthermore, we provide a quantitative comparison in Table 3. For different garment types such as jumpsuits and dresses, our proposed method achieves a higher level of accuracy, demonstrating its effectiveness in modeling deformations that are numerically close to physics-based simulations.

The ultimate aim of garment deformation is to enhance the user experience. To evaluate this, we conducted a user study involving animation videos of three different garments, with some qualitative results illustrated in Fig. 9. We recruited 35 participants for this study, including 12 with expertise in graphics and 23 from a computer science background without specific knowledge in computer graphics. Participants were asked to view all animations, which were presented in random order, and then rate each one based on realism and detail richness. The rating system used is a four-level scale ranging from 0 to 3, with each score being assigned uniquely. Partici-



**Fig. 11.** Example of the predicted garment with inconspicuous artifacts.

pants were allowed to replay the animation multiple times during the scoring process. As shown in Fig. 10, our method generally outperforms others in terms of detail richness. On the other hand, we discover GAPS also exhibits superior performance on the jumpsuit and the dress. It is important to note, however, that the GAPS model was trained specifically with these garments, a condition not replicated with the other methods during training.

## 6. Conclusions

We have developed a hybrid learning method that has proven capable of accurately predicting fine-level deformations in complex garments, enabling real-time inference. Our approach has improved the prediction of physics-aware garment deformations by leveraging the precision and reliability of supervised learning alongside the enhanced generalizability and exploration of diverse physical properties offered by unsupervised learning. Addressing the garment self-collision issue frequently highlighted in previous studies, we have thoroughly analyzed its origins and implemented a vertex repulsion-based constraint that has effectively minimized conspicuous intersections. Moreover, our framework features a logical processing flow that includes a multi-scale graph processing module for dynamic information handling. This setup, complete with structure-preserving pooling and unpooling techniques, has been essential in enhancing the propagation of information across the garment mesh, leading to the production of high-quality results. Several experiments demonstrate the effectiveness of our proposal.

There are also some limitations that could be addressed by follow-up works. While our method could alleviate the garment self-collision issue, it utilizes a soft constraint that may still fail under extreme variations in body movements and garment types that diverge from the training data. Such scenarios might require additional post-processing, potentially compromising efficiency. Furthermore, our approach is still unable to completely eliminate certain subtle self-collisions (as illustrated in Fig. 11); however, these artifacts are generally inconspicuous and do not affect the visual quality unless closely inspected frame by frame. Additionally, while our method is applied to the SMPL human body model and can be adapted for other basis human body models, it necessitates the body to have a fixed number of joints and vertices. Future research could explore a more general representation and processing mechanism for the human body information. Finally, although our method achieves real-time inference, we have not yet deployed it in VR/AR environments or conducted a detailed latency analysis in such settings. Integrating our approach into interactive engines remains an important direction for future work,

enabling practical applications and further evaluation of system efficiency in real-world scenarios.

#### CRediT authorship contribution statement

**Tianxing Li:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Methodology, Project administration, Writing – original draft. **Rui Shi:** Funding acquisition, Investigation, Software, Visualization, Writing – review and editing. **Qing Zhu:** Funding acquisition, Investigation, Resources, Validation, Writing – review and editing. **Takashi Kanai:** Formal analysis, Funding acquisition, Investigation, Validation, Writing – review and editing.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgements

This work has been partially supported by the National Natural Science Foundation of China (62402021, 62403017), Beijing Natural Science Foundation (4244088, 4232017), and Japan Society for the Promotion of Science (JSPS KAKENHI 25K15401).

#### References

- [1] N. Magnenat-Thalmann, R. Laperrière, D. Thalmann, Joint-dependent local deformations for hand animation and object grasping, in: Proc. Graph. Interface, 1989, pp. 26–33.
- [2] J. P. Lewis, C. Matt, F. Nickson, Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation, in: Proc. Annu. Conf. Comput. Graph. Interact. Tech., 2000, pp. 165–172. doi:10.1145/344779.344862.
- [3] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson, Physically based deformable models in computer graphics, Comput. Graph. Forum 25 (4) (2006) 809–836. doi:<https://doi.org/10.1111/j.1467-8659.2006.01000.x>.
- [4] D. Mongus, B. Repnik, M. Mernik, B. Žalik, A hybrid evolutionary algorithm for tuning a cloth-simulation model, Appl. Soft Comput. 12 (1) (2012) 266–273. doi:<https://doi.org/10.1016/j.asoc.2011.08.047>.
- [5] G. Cirio, J. Lopez-Moreno, D. Miraut, M. A. Otaduy, Yarn-level simulation of woven cloth, ACM Trans. Graph. 33 (6) (2014). doi:10.1145/2661229.2661279.
- [6] M. Zhang, T. Y. Wang, D. Ceylan, N. J. Mitra, Dynamic neural garments, ACM Trans. Graph. 40 (6) (2021). doi:10.1145/3478513.3480497.  
URL <https://doi.org/10.1145/3478513.3480497>
- [7] C. Patel, Z. Liao, G. Pons-Moll, TailorNet: Predicting clothing in 3d as a function of human pose, shape and garment style, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 7363–7373. doi:10.1109/CVPR42600.2020.00739.

- [8] X. Pan, J. Mai, X. Jiang, D. Tang, J. Li, T. Shao, K. Zhou, X. Jin, D. Manocha, Predicting loose-fitting garment deformations using bone-driven motion networks, in: ACM SIGGRAPH, 2022. doi:10.1145/3528233.3530709.
- [9] G. Han, M. Shi, T. Mao, X. Wang, D. Zhu, L. Gao, Self-Supervised Multi-Layer Garment Animation Generation Network, in: Pac. Grph., 2024. doi:10.2312/pg.20241307.
- [10] R. Vidaurre, I. Santesteban, E. Garces, D. Casas, Fully convolutional graph neural networks for parametric virtual try-on, *Comput. Graph. Forum* 39 (8) (2020) 145–156. doi:<https://doi.org/10.1111/cgf.14109>.
- [11] A. Grigorev, B. Thomaszewski, M. J. Black, O. Hilliges, HOOD: Hierarchical graphs for generalized modelling of clothing dynamics, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2023, pp. 16965–16974. doi:10.1109/CVPR52729.2023.01627.
- [12] T. Li, R. Shi, T. Kanai, Detail-aware deep clothing animations infused with multi-source attributes, *Comput. Graph. Forum* 42 (1) (2023) 231–244. doi:<https://doi.org/10.1111/cgf.14651>.
- [13] H. Bertiche, M. Madadi, S. Escalera, PBNS: Physically based neural simulation for unsupervised garment pose space deformation, *ACM Trans. Graph.* 40 (6) (2021). doi:10.1145/3478513.3480479.
- [14] I. Santesteban, M. A. Otaduy, D. Casas, SNUG: Self-supervised neural dynamic garments, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 8130–8140. doi:10.1109/CVPR52688.2022.00797.
- [15] H. Bertiche, M. Madadi, S. Escalera, Neural cloth simulation, *ACM Trans. Graph.* 41 (6) (2022). doi:10.1145/3550454.3555491.
- [16] T. Li, R. Shi, Z. Li, T. Kanai, Q. Zhu, Efficient deformation learning of varied garments with a structure-preserving multilevel framework, in: Proc. Symp. Interactive 3D Graph. Games, 2024. doi:10.1145/3651286.
- [17] K.-J. Choi, H.-S. Ko, Stable but responsive cloth, *ACM Trans. Graph.* 21 (3) (2002) 604–611. doi:10.1145/566654.566624.
- [18] R. Narain, A. Samii, J. F. O’Brien, Adaptive anisotropic remeshing for cloth simulation, *ACM Trans. Graph.* 31 (6) (2012). doi:10.1145/2366145.2366171.
- [19] C. Yuan, H. Shi, L. Lan, Y. Qiu, C. Yuksel, H. Wang, C. Jiang, K. Wu, Y. Yang, Volumetric homogenization for knitwear simulation, *ACM Trans. Graph.* 43 (6) (2024). doi:10.1145/3687911.
- [20] C. Jiang, T. Gast, J. Teran, Anisotropic elastoplasticity for cloth, knit and hair frictional contact, *ACM Trans. Graph.* 36 (4) (2017). doi:10.1145/3072959.3073623.
- [21] J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, L. Boissieux, An implicit frictional contact solver for adaptive cloth simulation, *ACM Trans. Graph.* 37 (4) (2018). doi:10.1145/3197517.3201308.

- [22] M. Tang, H. Wang, L. Tang, R. Tong, D. Manocha, CAMA: Contact-aware matrix assembly with unified collision handling for gpu-based cloth simulation, *Comput. Graph. Forum* 35 (2) (2016) 511–521. doi:<https://doi.org/10.1111/cgf.12851>.
- [23] L. Wu, B. Wu, Y. Yang, H. Wang, A safe and fast repulsion method for gpu-based cloth self collisions, *ACM Trans. Graph.* 40 (1) (2020). doi:10.1145/3430025.
- [24] G. Cirio, J. Lopez-Moreno, D. Miraut, M. A. Otaduy, Yarn-level simulation of woven cloth, *ACM Trans. Graph.* 33 (6) (2014). doi:10.1145/2661229.2661279.
- [25] T. Vassilev, S. Bernhard, C. Yiorgos, Fast cloth animation on walking avatars, *Comput. Graph. Forum* 20 (3) (2001) 260–267. doi:<https://doi.org/10.1111/1467-8659.00518>.
- [26] Z. Chen, H.-Y. Chen, D. M. Kaufman, M. Skouras, E. Vouga, Fine wrinkling on coarsely meshed thin shells, *ACM Trans. Graph.* 40 (5) (2021). doi:10.1145/3462758.
- [27] H. Wang, GPU-based simulation of cloth wrinkles at submillimeter levels, *ACM Trans. Graph.* 40 (4) (2021). doi:10.1145/3450626.3459787.
- [28] L. Sigal, M. Mahler, S. Diaz, K. McIntosh, E. Carter, T. Richards, J. Hodgins, A perceptual control space for garment simulation, *ACM Trans. Graph.* 34 (4) (2015). doi:10.1145/2766971.
- [29] S. Yang, Z. Pan, T. Amert, K. Wang, L. Yu, T. Berg, M. C. Lin, Physics-inspired garment recovery from a single-view image, *ACM Trans. Graph.* 37 (5) (2018). doi:10.1145/3026479.
- [30] M.-H. Jeong, D.-H. Han, H.-S. Ko, Garment capture from a photograph, *Comput. Animat. Virtual Worlds* 26 (2015) 291–300. doi:<https://doi.org/10.1002/cav.1653>.
- [31] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, C. Theobalt, Video-based reconstruction of animatable human characters, in: *ACM SIGGRAPH Asia*, 2010. doi:10.1145/1866158.1866161.
- [32] Z. Xu, S. Peng, C. Geng, L. Mou, Z. Yan, J. Sun, H. Bao, X. Zhou, Relightable and animatable neural avatar from sparse-view video, in: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 990–1000. doi:10.1109/CVPR52733.2024.00100.
- [33] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, H. Bao, Bcnet: Learning body and cloth shape from a single image, in: *Proc. Eur. Conf. Comput. Vis.*, Vol. 12365, 2020, pp. 18–35. doi:10.1007/978-3-030-58565-5\_2.
- [34] M. Shi, W. Feng, L. Gao, D. Zhu, Generating diverse clothed 3d human animations via a generative model, *Comput. Vis. Media* 10 (2) (2024) 261–277. doi:10.1007/S41095-022-0324-2.
- [35] I. Santesteban, M. A. Otaduy, D. Casas, Learning-based animation of clothing for virtual try-on, *Comput. Graph. Forum* 38 (2) (2019) 355–366. doi:<https://doi.org/10.1111/cgf.13643>.
- [36] T. Y. Wang, D. Ceylan, J. Popović, N. J. Mitra, Learning a shared shape space for multi-modal garment design, *ACM Trans. Graph.* 37 (6) (2018). doi:10.1145/3272127.3275074.

- [37] G. Tiwari, B. L. Bhatnagar, T. Tung, G. Pons-Moll, SIZER: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing, in: Proc. Eur. Conf. Comput. Vis., Vol. 12348, 2020, pp. 1–18. doi:10.1007/978-3-030-58580-8\_1.
- [38] R. Chen, L. Chen, S. Parashar, GAPS: Geometry-aware, physics-based, self-supervised neural garment draping, in: Int. Conf. on 3D Vis., 2024, pp. 116–125. doi:10.1109/3DV62453.2024.00059.
- [39] Y. Li, M. Tang, Y. bo Yang, Z. Huang, R. Tong, S. Yang, Y. Li, D. Manocha, N-Cloth: Predicting 3d cloth deformation with mesh-based networks, Comput. Graph. Forum 41 (2) (2022) 547–558. doi:<https://doi.org/10.1111/cgf.14493>.
- [40] Y. Li, M. Tang, X. R. Chen, Y. Yang, R. Tong, B. An, S. Yang, Y. Li, Q. Kou, D-cloth: Skinning-based cloth dynamic prediction with a three-stage network, Comput. Graph. Forum 42 (7) (2023). doi:10.1111/CGF.14937.
- [41] Y. Li, M. Tang, Y. Yang, R. Tong, S. Yang, Y. Li, B. An, Q. Kou, CTSN: predicting cloth deformation for skeleton-based characters with a two-stream skinning network, Comput. Vis. Media 10 (3) (2024) 471–485. doi:10.1007/S41095-023-0344-6.
- [42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proc. Int. Conf. Learn. Representations, 2018.
- [43] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, M. J. Black, SMPL: A skinned multi-person linear model, ACM Trans. Graph. 34 (6) (2015). doi:10.1145/2816795.2818013.
- [44] E. Gundogdu, V. Constantin, S. Parashar, A. Seifoddini, M. Dang, M. Salzmann, P. Fua, GarNet++: Improving fast and accurate static 3D cloth draping by curvature loss, IEEE Trans. Pattern Anal. Mach. Intell. 44 (01) (2022) 181–195. doi:10.1109/TPAMI.2020.3010886.
- [45] M. Zhang, D. Ceylan, N. J. Mitra, Motion guided deep dynamic 3d garments, ACM Trans. Graph. 41 (6) (2022). doi:10.1145/3550454.3555485.
- [46] C. Du, F. Yu, M. Jiang, A. Hua, Y. Zhao, X. Wei, T. Peng, X. Hu, High fidelity virtual try-on network via semantic adaptation and distributed componentization, Comput. Vis. Media 8 (4) (2022) 649–663. doi:10.1007/S41095-021-0264-2.
- [47] X. Zhang, J. Chen, L. Ma, K. Zhang, A virtual try-on network with arm region preservation, Appl. Soft Comput. 175 (2025) 112960. doi:<https://doi.org/10.1016/j.asoc.2025.112960>.
- [48] A. M. Aladdin, T. A. Rashid, LEO: lagrange elementary optimization, Neural Comput. Appl. 37 (19) (2025) 14365–14397. doi:10.1007/S00521-025-11225-2.
- [49] A. A. H. Amin, A. M. Aladdin, D. O. Hasan, S. R. Mohammed-Taha, T. A. Rashid, Enhancing algorithm selection through comprehensive performance evaluation: Statistical analysis of stochastic algorithms, Comput. 11 (11) (2023) 231. doi:10.3390/COMPUTATION11110231.

- [50] S. R. Mohammed-Taha, A. M. Aladdin, A. I. Mustafa, D. O. Hasan, R. K. Muhammed, T. A. Rashid, Analyzing the impact of the pandemic on insomnia prediction using machine learning classifiers across demographic groups, *Netw. Model. Anal. Health Informatics Bioinform.* 14 (1) (2025) 51. doi:10.1007/S13721-025-00552-Y.
- [51] D. Lee, D. Lee, K. Kim, Self-growth learning-based machine scheduler to minimize setup time and tardiness in OLED display semiconductor manufacturing, *Appl. Soft Comput.* 145 (2023) 110600. doi:10.1016/J.ASOC.2023.110600.
- [52] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, J. Leskovec, Hierarchical graph representation learning with differentiable pooling, in: *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.
- [53] H. Gao, S. Ji, Graph u-nets, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (9) (2022) 4948–4960. doi:10.1109/TPAMI.2021.3081010.
- [54] J. Lee, I. Lee, J. Kang, Self-attention graph pooling, in: *Proc. Int. Conf. Mach. Learn.*, Vol. 97, 2019, pp. 3734–3743.
- [55] E. Ranjan, S. Sanyal, P. P. Talukdar, ASAP: Adaptive structure aware pooling for learning hierarchical graph representations, in: *Proc. AAAI Conf. Artif. Intell.*, 2019.
- [56] M. Lino, S. Fotiadis, A. A. Bharath, C. D. Cantwell, Multi-scale rotation-equivariant graph neural networks for unsteady eulerian fluid dynamics, *Phys. Fluids* 34 (8) (2022) 087110. doi:10.1063/5.0097679.
- [57] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the nystrom method, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2) (2004) 214–225. doi:10.1109/TPAMI.2004.1262185.
- [58] H. Bertiche, M. Madadi, E. Tyson, S. Escalera, DeePSD: Automatic deep skinning and pose space deformation for 3D garment animation, in: *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 5471–5480.
- [59] D. Baraff, A. P. Witkin, Large steps in cloth simulation, in: *ACM SIGGRAPH*, 1998, pp. 43–54. doi:10.1145/280814.280821.
- [60] H. Bertiche, M. Madadi, S. Escalera, CLOTH3D: clothed 3d humans, in: *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 344–359.
- [61] Carnegie-Mellon, CMU graphics lab motion capture database, <http://mocap.cs.cmu.edu/>, accessed: 2024 (2010).
- [62] R. J. Aumann, L. S. Shapley, *Values of non-atomic games*, Princeton University Press, 1974.