Efficient Deformation Learning of Varied Garments with a Structure-Preserving Multilevel Framework

TIANXING LI, Beijing University of Technology, China
RUI SHI*, Beijing University of Technology, China
ZIHUI LI, Beijing University of Technology, China
TAKASHI KANAI, The University of Tokyo, Japan
QING ZHU, Beijing University of Technology, China

Due to the highly nonlinear behavior of clothing, modelling fine-scale garment deformation on arbitrary meshes under varied conditions within a unified network poses a significant challenge. Existing methods often compromise on either model generalization, deformation quality, or runtime speed, making them less suitable for real-world applications. To address it, we propose to incorporate multi-source graph construction and pooling to achieve a novel graph learning scheme. We first introduce methods for extracting cues from different deformation correlations and transform the garment mesh into a comprehensive graph enriched with deformation-related information. To enhance the learning capability and generalizability of the model, we present structure-preserving pooling and unpooling strategies for the mesh deformation task, thereby improving information propagation across the mesh and enhancing the realism of deformation. Lastly, we conduct an attribution analysis and visualize the contribution of various vertices in the graph to the output, providing insights into the deformation behavior. The experimental results demonstrate superior performance against state-of-the-art methods.

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

Additional Key Words and Phrases: Clothing deformation, Nerual network, Graph pooling, Attribution analysis

ACM Reference Format:
Tianxing Li, Rui Shi, Zihui Li, Takashi Kanai, and Qing Zhu. 2024. Efficient Deformation Learning of Varied Garments with a Structure-Preserving Multilevel Framework. *J. ACM* 37, 4, Article 111 (August 2024), 18 pages. https://doi.org/xxxxx.xxxxx

1 INTRODUCTION

The development of digital garments with accurate behaviors for dressing avatars is a vital focus in computer graphics research, applicable to industries from film and gaming to virtual try-ons and the metaverse. However, traditional methods often confront a predicament between deformation quality and generation efficiency. Methods such as linear blend skinning [Magnenat-Thalmann et al. 1989] and pose space deformation [Lewis et al. 2000] are straightforward to implement and are capable of producing garment deformations quickly, thereby enabling clothing animations to

*Corresponding Author

Authors' addresses: Tianxing Li, litianxing@bjut.edu.cn, Beijing University of Technology, Beijing, China; Rui Shi, Beijing University of Technology, Beijing, China, ruishi@bjut.edu.cn; Zihui Li, Beijing University of Technology, Beijing, China, lizihui@emails.bjut.edu.cn; Takashi Kanai, The University of Tokyo, Tokyo, Japan, kanait@acm.org; Qing Zhu, Beijing University of Technology, Beijing, China, ccgszq@bjut.edu.cn.

0004-5411/2024/8-ART111 $15.00
https://doi.org/xxxxx.xxxxx

run at interactive speeds. Nonetheless, these techniques often compromise on realism, leading to a lack of fine detail and dynamic behavior. In contrast, physics-based simulations [Cirio et al. 2014a; Nealen et al. 2006] offer high-quality results by numerically resolving each step of geometric changes in the garment mesh. However, these methods are computationally intensive and can be prohibitive, especially for time-sensitive or computationally resource-constrained applications.

In recent years, machine learning has demonstrated significant potential in enhancing performance and fidelity in complex tasks. The use of neural networks in the realm of clothing deformation has similarly yielded successful results, effectively approximating clothing behavior through low-dimensional parametrization of body shape and pose. Despite this, the highly specific nature of the trained model to a particular garment creates challenges when applying it to other garments with arbitrary mesh topology and varying numbers of vertices [Bertiche et al. 2021, 2022; Patel et al. 2020]. This lack of generalizability significantly restricts the real-world applicability of these methods, given that most scenarios demand a diverse range of garments.

As a consequence, the utilization of graph neural networks (GNNs) in clothing deformation has gained interest in recent studies [Grigorev et al. 2023; Li et al. 2023a; Vidaurre et al. 2020]. This is primarily due to the inherent ability to generalize, notably allowing the mesh structure to be agnostic. However, vanilla graph learning-based approaches face substantial technical challenges due to the highly individualized variations in mesh deformations across diverse garments and the extensive scale of data: (1) **model learning difficulty**: as the number and variability of mesh graphs increase, the model needs to deepen to manage complex features, yet issues with model convergence and gradient vanishing problem. (2) **unnatural outputs**: deep GNNs can be likened to low-pass filters where graph updates between layers primarily originate from the local feature aggregation across neighborhood vertices. This leads to difficulties in long-distance information transition, resulting in poor interactions among vertices in simulated garments.

Motivated by existing challenges, we propose a novel multilevel framework to handle irregular domains with arbitrary garment meshes based on generalized attention mechanism (as depicted in Fig. 1). In particular, we initially analyze and categorize the key factors influencing garment deformation, and then design distinct network components to handle multi-source garment behavior-related information. This process results in the creation of graph representations rich in deformation cues. Then, to ensure effective propagation and exchange of information among graph nodes, we propose straightforward yet effective pooling and unpooling strategies specifically tailored for the clothing deformation task. In alignment with the standard 3D animation pipeline, our network is structured to learn a nonlinear mapping from an input space, which includes garment-body features, to garment skinning weights and blend shapes. Our approach offers a simple implementation process and also ensures compatibility with any graphics engine. Our contributions are as follows:

- We devise a set of strategies for generating descriptions centered on deformation-related information, aiding in the construction of garment graph representations enriched with multi-source cues. For the body and the garment global information, we design distinct components to extract their high-dimensional features according to their respective characteristics. These features, serving as latent cues, are then combined with the local features of the garment, resulting in an integrated graph representation. This representation enables easy application of the model to scenes featuring diverse objects.
- We present structure-preserving pooling and unpooling methods for graph learning-based approximation of garment deformation. Unlike other multiscale GNNs, our method provide a robust solution that both retains the mesh connectivity of the overall structure during the mesh coarsening process, and considers the neighbor information of the original graph nodes

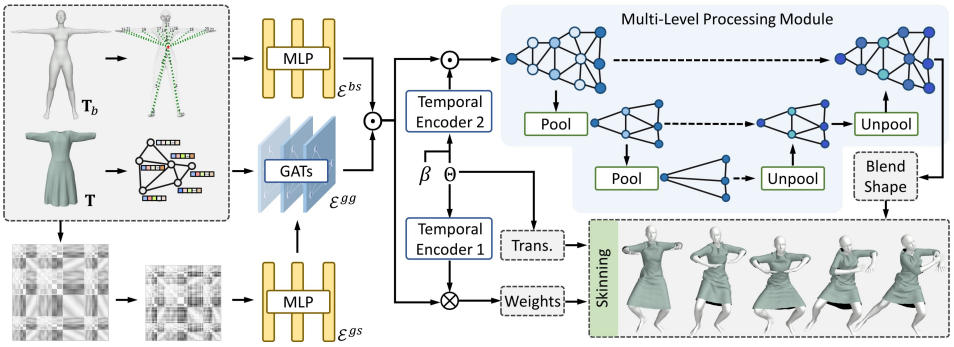

Fig. 1. The overview of our deformation network. The network operates as an end-to-end system, taking input in the form of garment, body, and motion sequences to generate deformation results.

when refining back to a fine mesh. The method facilitates the diffusion and aggregation of information within the mesh, leading to improved learning ability and deformation quality.

- We conduct an attribution analysis for evaluation aimed at enhancing the explainability of our approach. With the use of a gradient-based attribution method, we highlight how the proposed pooling and unpooling strategies influence network decisions and directly affect the behavior of garment mesh nodes.

We also evaluate our method on multiple garments and human bodies, achieving reasonable deformation predictions at an average rate of 402.1 frames per second (fps). In comparison to state-of-the-art learning-based approaches, our method excels in establishing a well-rounded balance between model generalization, deformation quality, and efficiency, thereby demonstrating superior overall performance.

2 RELATED WORKS

In this section, we discuss existing garment deformation approaches, by classifying them into physics-based simulations and learning-based methods.

Physics-based simulations (PBS) are widely recognized for their ability to achieve a high degree of realism in deformation effects [Choi and Ko 2002; Narain et al. 2012]. They leverage the discretizations of classical mechanics to accurately simulate the deformation process of cloth. Despite their impressive capability to generate realistically detailed deformation outcomes, they inherently come with substantial computational demands, require extensive computational resources, and pose significant challenges in reaching interactive speeds [Jiang et al. 2017; Li et al. 2018]. To enhance the efficiency of PBS, a considerable amount of research has been conducted. Approaches include leveraging the parallel computational capabilities of modern GPUs [Cirio et al. 2014b; Tang et al. 2016; Wu et al. 2020], simplifying and focusing on constrained scenarios [Vassilev et al. 2001], and adding intricate high-frequency wrinkles to low-frequency meshes [Chen et al. 2021; Wang 2021]. Nevertheless, even with these advancements, these approaches remain inadequate when faced with limited computational capacity. Despite research attempts to introduce perceptual control [Sigal et al. 2015] and the automatic inference of physical parameters [Jeong et al. 2015; Stoll et al. 2010; Yang et al. 2018], these methods are restricted to function optimally only in controlled environments and still require the implementation of PBS. Consequently, these approaches continue to demand substantial time for upfront parameter tuning and demand a high level of expertise in parameter adjustment.

Learning-based methods aim at learning a function that directly yields the anticipated deformation for any specified input [Li et al. 2023b; Pan et al. 2022; Santesteban et al. 2022b, 2021]. These approaches have seen increased adoption in the realm of clothing animation in recent years, primarily due to their superior efficiency and enhanced automation compared to traditional physics-based simulations. Building on the foundational work of pose space deformation [Lewis et al. 2000], several studies [Hahn et al. 2014; Patel et al. 2020; Santesteban et al. 2019; Tiwari et al. 2020; Wang et al. 2018] have suggested learning garment deformations from various parameters such as pose, shape, or garment size. Although these models exhibit rapid performance, they struggle to predict plausible folds when dealing with loose garments. To enhance the reconstruction of garment appearance details, researchers [Bertiche et al. 2021, 2022; Santesteban et al. 2022a] recast the physics-based simulation as an optimization problem, incorporating a set of physics-based loss terms. These methods also introduce unsupervised strategies, eliminating the need for ground truth data during the training process. More recently, method has been developed that can effectively generate deformations for loose-fitting dresses [Pan et al. 2022]. However, a shared limitation of all the aforementioned methods is their inability to generalize to garments with different mesh structures.

To mitigate the model generalization issue, recent studies have shifted focus to the use of graph neural networks (GNNs). This is largely due to their remarkable ability to process 3D data and their independence from requiring upfront knowledge of the mesh graph structure. For instance, Li *et al.* [Li et al. 2020, 2021] apply graph attention network (GAT) [Veličković et al. 2018] to predict nonlinear deformations across a variety of articulated characters. However, these methods still face challenges in accurately reproducing complicated clothing wrinkles. In response to this challenge, the authors subsequently propose a unified graph-attention-based detail-aware network (DANet) for garment deformations [Li et al. 2023a], with the garments worn on top of the parametrized SMPL body [Loper et al. 2015]. Meanwhile, other works use different architectures such as GraphUNet-based [Grigorev et al. 2023; Vidaurre et al. 2020] and PointNet-based frameworks [Gundogdu et al. 2022] to perform the task of cloth deformation. However, these methods do not entirely solve the generalization problem: they either struggle with variations in topology, or face difficulties in effectively capturing and leveraging global information derived from the available data to produce satisfactory results for new garments or poses.

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 [Ying et al. 2018], TopKPool [Gao and Ji 2022], SAGPool [Lee et al. 2019], ASAP [Ranjan et al. 2019], and rasterization-based pooling methods [Lino et al. 2021, 2022]. Despite their innovations, these methods contend with common obstacles such as pooling inefficiencies and the preservation of stable mesh structures-critical for maintaining the integrity of 3D models and other graph-based analysis.

3 METHODOLOGY

3.1 Garment Deformation Model

In computer animation, the standard deformation pipeline, which leverages skinning weights and blend shapes to calculate a garment deformation in a specific pose, is a common practice. Known for its simplicity and high compatibility, our approach also adopts this pipeline. In particular, we begin with an initial garment mesh template $\mathbf{T} \in \mathbb{R}^{N\times 3}$ with arbitrary number of vertices N, a basis SMPL [Loper et al. 2015] human body $\mathbf{T}_b \in \mathbb{R}^{N_b\times 3}$ with N_b number of vertices and shape

parameter β, and a sequence of pose histories $\Theta^t = \{\theta^t, \theta^{t-1}, ..., \theta^{t-m+1}\}$ from $(t-m+1)$-th frame to the current frame t, our goal is to learn a garment deformation network capable of establishing an accurate mapping from these variables to garment temporal skinning weights and blend shapes. This mapping can be represented as follows:

$$\text{Network} : \{\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta\} \rightarrow \{\mathcal{W}^t, B^t\}, \tag{1}$$

where $\mathcal{W}^t \in \mathbb{R}^{N\times S}$ denotes the garment skinning weights and S is the joint number of the body skeleton. $B \in \mathbb{R}^{N\times 3}$ represents the blend shape. Next, by applying a skinning function W, the garment deformation at t-th frame M^t can be obtained:

$$M^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta) = W(T^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta), J(\beta), \theta^t, \mathcal{W}^t(\mathbf{T}, \mathbf{T}_b, \Theta^t)), \tag{2}$$

$$T^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta) = \mathbf{T} + B^t(\mathbf{T}, \mathbf{T}_b, \Theta^t, \beta), \tag{3}$$

where the deformed garment mesh T^t in rest pose is calculated by adding the predicted blend shape $B^t(\cdot)$ to the mesh template $\mathbf{T}$. Using the learned skinning weights $\mathcal{W}^t$ and the body joint location $J(\cdot)$, the skinning function $W(\cdot)$ (*e.g.*, linear blend skinning) deforms the unposed mesh T^t into the deformed mesh M^t under the current state t.

In real-world applications, clothing comes in various styles, encompassing not just tight-fitting garments but also loose-fitting ones. Consequently, we cannot simply assume that the garment mesh M^t deforms in the same way as the body, as in general skinning algorithms. To ensure a natural outcome for M^t, the skinning weights $\mathcal{W}^t$ and blend shapes B^t in Eq. (2) and Eq. (3) need to satisfy the following conditions. First, the skinning weights must accurately capture the dynamic impacts of temporal skeletal transformations on the vertices of different garments during the movement. Second, the blend shapes need to be adjusted non-linearly, factoring in multiple sources of features from both the human body and the garment itself, along with the motion state, in order to ensure coherent detailed deformation. In the following subsection, we will provide a detailed explanation on how to integrate various types of information to ensure that $\mathcal{W}^t$ and B^t fulfill the aforementioned conditions.

3.2 Multi-Source Information Processing

Garment graph information. Our objects of deformation are 3D garments, which are inherently complex, showcasing a wide range of shapes and topologies. Moreover, the vertex-to-vertex interactions within a given garment exhibit flexible changes corresponding to the deformation. Consequently, to represent such data in a compact and efficient manner, we opt to construct a graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ for the garment mesh. Here, $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ is the feature set of all N vertices, and $\mathcal{E}$ represents the mesh edges, denoting the connectivity between vertices. To make the vertices distinguishable, we assign features to each vertex, including vertex position, normal, and distance to all body joints, resulting in the feature vector x_i.

After constructing the garment graph, we employ a garment graph encoder $\mathcal{E}^{gg} : \mathbb{R}^{N\times F_0} \rightarrow \mathbb{R}^{N\times F\times K}$ to transform input graph node features $X = [x_1, x_2, ..., x_N]^\top$ into high-level latent representations. To handle the irregularities inherent in the graph structure and prioritize significant node information, we introduce a mask self-attention block in $\mathcal{E}^{gg}$, drawing inspiration from the GAT. Specifically, the feature transformation in one attention block can be expressed as:

$$x_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k H^k x_i\right), \tag{4}$$

where x_i' denotes the transformed feature of node i, σ is the nonlinear activation function, K is the number of attention heads (set as $K = S$, the number of body joints), α_{ij}^k is the attention

coefficient between nodes i and its neighborhood j in the k-th attention head, H^k is the learned linear transformation for the k-th attention head, and x_i is the original feature of node i. The multi-head features can be concatenated or summed in practice. After processed by several attention blocks, the graph can be encoded as $\mathcal{G}^{gg} := \mathcal{E}^{gg}(\mathcal{G})$ with node features X^{gg}.

Garment style information. While we incorporate local information for each vertex within the garment graph, it is important to note that the global context of the garment is also pivotal in effectively modelling garment deformation. For instance, garment dresses or jumpsuits exhibit significantly different styles. It is essential to characterize their shapes holistically to guide their individual deformation behavior within a unified model effectively. To achieve this, we construct an $N \times N$ affinity matrix G for the garment mesh template $\mathbf{T}$. In this matrix, G_{ij} signifies the affinity between the i-th and j-th vertices, which we choose to represent using geodesic distance. The distances inherently reflect the shape of the garment and assist in establishing semantically meaningful relationships. For instance, two vertices situated on the same sleeve of a shirt are likely to share a closer relationship (despite potentially being distant in Euclidean space) than a vertex on a sleeve and another on the shirt's body. To improve the efficiency without loss of informative expressiveness, we apply the Nyström approximation [Fowlkes et al. 2004] to generate a low-rank approximation $\tilde{G}^{nys} = UW^{+}U^{\top} \in \mathbb{R}^{N^* \times N^*}$ of the original G, where $N^* \ll N$. $U \in \mathbb{R}^{N \times N^*}$ is a matrix consisting of N^* columns sampled from G and $W \in \mathbb{R}^{N^* \times N^*}$ is the matrix consisting of the intersection of these columns with the corresponding rows of G.

As in the bottom left of Fig. 1, we visualize the matrix G and its approximation $\tilde{G}^{nys}$ according to the numerical magnitude. We use the eigenvalues of $\tilde{G}^{nys}$ as global descriptors to characterize the shape of a garment. Then, these eigenvalues are forwarded into a garment style encoder $\mathcal{E}^{gs} : \mathbb{R}^{N^*} \rightarrow \mathbb{R}^{\times F}$ to generate style-related latent cues $C^{gs} := \mathcal{E}^{gs}(\tilde{G}^{nys})$. These cues are multiplied with X^{gg} and fed into another mask self-attention block to yield $\hat{X}^{gg} \in \mathbb{R}^{N \times F \times S}$. The resulting feature is subsequently used to integrate with body shape information.

Body shape information. Another crucial factor influencing garment deformation is the underlying body shape, as different body shapes induce varying global and local deformation behaviors in garments. For example, a garment worn on a larger body with a tight fit tends to conform more closely to the body's movement, resulting in denser detail folds. Conversely, a garment on a slimmer body often displays more dynamic behavior, typically characterized by wider folds. To characterize specific dimensions and proportions of different body parts, we introduce a joint-vertex distance feature $D \in \mathbb{R}^{N_b \times S}$ for body mesh $\mathbf{T}_b$ to describe its shape.

Subsequently, we process these shape features using a body shape encoder $\mathcal{E}^{bs} : \mathbb{R}^{N_b \times S} \rightarrow \mathbb{R}^{F \times S}$. Composed of multilayer perceptrons (MLPs), this encoder transforms the features into body-related latent cues, denoted as $C^{bs} := \mathcal{E}^{bs}(D)$.

Skinning weights and blend shapes generation. Incorporating deformation related cues into the garment graph is a crucial step in achieving a more accurate representation of garment behavior. To this end, we combine information from body and garment together:

$$X^{fuse} = \hat{X}^{gg} \odot C^{bs}, \tag{5}$$

where $\odot$ denotes element-wise multiplication. After integration, $X^{fuse} \in \mathbb{R}^{N \times F \times S}$ serves as the feature matrix of the graph $\mathcal{G}^{fuse}$, providing a more comprehensive representation of the garment deformation association information. Note that for each garment-body pair, this integrated graph needs to be generated only once initially.

Temporal information related to motion significantly influences the dynamics of a garment, particularly for relatively loose items like dresses and wide-legged trousers. To ensure a continuous and natural effect, dynamic skinning weights and blend shapes must be assigned to each state. Specifically, in the generation of skinning weights, given a motion sequence $\Theta^t = \{\theta^t, \theta^{t-1}, ..., \theta^{t-m+1}\}$

where each θ encompasses the axis-angle of each joint and the translation of the root relative to the preceding frame, we utilize a temporal encoder $\mathcal{E}^{t1} : \mathbb{R}^{3(S+1)\times m} \rightarrow \mathbb{R}^F$ composed of long short-term memory (LSTM) to extract the motion state features. These features are then combined with the integrated graph features to obtain the current skinning weights $\mathcal{W}^t$:

$$\mathcal{W}^t = X^{fuse} \otimes \mathcal{E}^{t1}(\Theta^t), \tag{6}$$

where $\otimes$ is the multiplication operator. The process of generating blend shapes is more complex compared to that of skinning weights, as it necessitates compensation for personalized detail folds. To accommodate this, we initially process the body shape β and motion Θ^t further in order to compute the velocity states $\mathbf{V}^t$ of all body vertices. The body state features are then extracted by another temporal encoder $\mathcal{E}^{t2} : \mathbb{R}^{3N_b\times m} \rightarrow \mathbb{R}^{FS}$ and combined with $X^{fuse'}$ generated by flattening X^{fuse} to form motion-related integrated graph features $\tilde{X}^{fuse}$. The corresponding graph $\tilde{\mathcal{G}}^{fuse}$ is finally input into the multilevel processing module $\mathcal{M}^{mp}$ to generate the blend shapes:

$$\tilde{X}^{fuse} = X^{fuse'} \odot \mathcal{E}^{t2}(\mathbf{V}^t), \tag{7}$$

$$B^t := \mathcal{M}^{mp}\left(\tilde{\mathcal{G}}^{fuse}\right). \tag{8}$$

To efficiently learn the details of mesh deformation, we incorporate the multilevel structure used in $\mathcal{M}^{mp}$, including both pooling and unpooling strategies. We will elaborate on this aspect in the subsequent subsection.

3.3 Structure-Preserving Pooling

Garment deformation should be able to represent the propagation of information within the mesh and their subsequent impact on the deformation details. Consequently, for a deformation model, the influence of a vertex should extend to as many neighboring vertices as possible during the feature propagation to avoid the "locality problem." Graph pooling naturally aids in the rapid diffusion of vertex information. However, unlike other graph machine learning tasks, garment graph pooling should have the ability to preserve the connectivity of the garment's overall structure, which is the basis for achieving neural simulation on clothing mesh. The corresponding unpooling should fully consider the surrounding vertex information to return the fine graph. Simultaneously, given that vertices could spatially come into contact due to bending and shearing during the deformation process, graph pooling should avoid being affected by Euclidean spatial proximity. Based on these considerations, we use the garment graph information itself to design the pooling strategy.

Our initial inspiration stems from the preservation of second-order connections in a directed acyclic graph. As shown in the pooling part of Fig. 1, every alternate depth's pooling eliminates current first-order neighbor vertices. To achieve a partition on a garment mesh, we initially determine a seed vertex e^{ecc} according to the eccentricity of the garment graph $\mathcal{G}^0$ in rest pose and calculate the geodesic distances d from the seed to all other vertices. It is important to clarify that the term "geodesic distance" in this context is drawn from graph theory and signifies the shortest path connecting two vertices, measured by the count of edges along the minimal connecting path. Following this, we traverse the mesh, discarding vertices $\mathcal{X}^r$ that exhibit an odd geodesic distance, thereby obtaining the vertex set $\mathcal{X}^c$ of the pooled graph. 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. The detailed procedure of graph pooling is shown in Algorithm 1 of Appendix A.

For unpooling, we find that weighted filling of vertex features, based on the neighbor information A of the fine graph, effectively eliminates wave-like deformation artifacts. The details of our graph unpooling are provided in Algorithm 2 of Appendix A. For clarity, we append a superscript to the

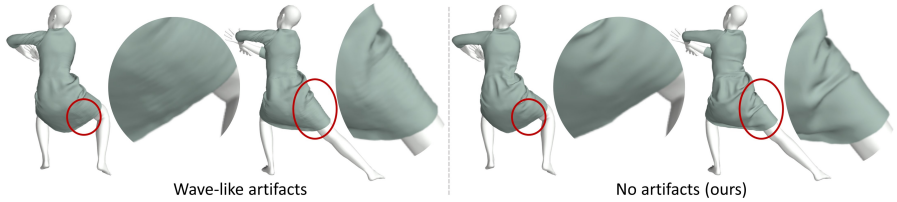

Fig. 2. The wave-like artifacts when using directly using vertex features before pooling for unpooling feature transition.

vertex feature variables in the unpooling algorithm. Geometrically, $A_{ij} = 1$ indicates the existence of edge (i, j). Our method begins by determining the neighbor relationships $\mathcal{N}^{r0}$ and $\mathcal{N}^{c0}$, which are obtained through the set of *removed* vertices and the vertex set of the *coarse* graph. Following this, we calculate the weight $\hat{w}_{ij}$ and proceed to directly construct the coarse vertex features into the fine graph. Finally, we normalize $\hat{w}_{ij}$ and use edge weight w_{ij} to weight the neighboring vertices of the removed vertex to reconstruct its features.

The positive effects of using graph pooling on character and garment deformation models have been verified [Li et al. 2021; Vidaurre et al. 2020]. More recently, hierarchical graphs are designed to achieve diverse garment deformation with one network [Grigorev et al. 2023], that their idea aligns with our approach. Though both methods leverage this idea, the modes of information transition we adopt differ in important ways. They discard the connectivity of vertices in the original graph after pooling operation, and directly use vertex features before pooling to populate vertex features during unpooling operation. In contrast, our method maintains the fine graph's connectivity within the coarse graph, leveraging neighborhood information during unpooling for vertex feature returning. Although it is challenging to theoretically ascertain which mode prevails, when directly applied to generate deformation using the skinning weights and blend shape pipeline, a standard in almost all graphics engines, our method successfully address the wave-like artifacts in the results compared with other transition modes, as shown in Fig. 2. Such artifacts typically appear in flatter clothing areas, but not all poses could induce artifacts. Accounting for neighboring vertex information during transition can effectively address this problem. After establishing the strategy of pooling and unpooling, we follow the GraphUNet [Gao and Ji 2022] fashion to construct our multi-resolution processing module $\mathcal{M}^{mp}$ for generating blend shapes B^t. Consequently, with the skinning weights and blend shapes, we are able to compute the final garment deformation M^t.

4 EXPERIMENTS

For the dataset, we collect various types of garments from CLOTH3D [Bertiche et al. 2020], human body from SMPL [Loper et al. 2015], and continuous motions from CMU Mocap dataset [Carnegie-Mellon 2010]. 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 10 garments with randomly generated body shapes, with around 8,000 poses. There is no overlap among these datasets, ensuring their independence. For further details on the network architecture and implementation, please refer to the supplemental materials.

4.1 Evaluation on Multilevel Processing

GAT blocks allow for the continuous aggregation of vertex features from first-order neighbors. The field within which a single vertex can exert influence is equivalent to the depth of the GAT-based

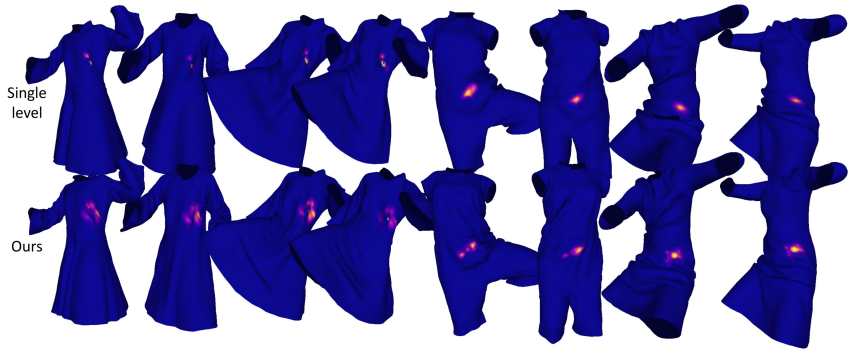

Fig. 3. The attribution results of the single-level network and our multilevel one. The deformation result in our network always receive a broader impact from ambient vertices.

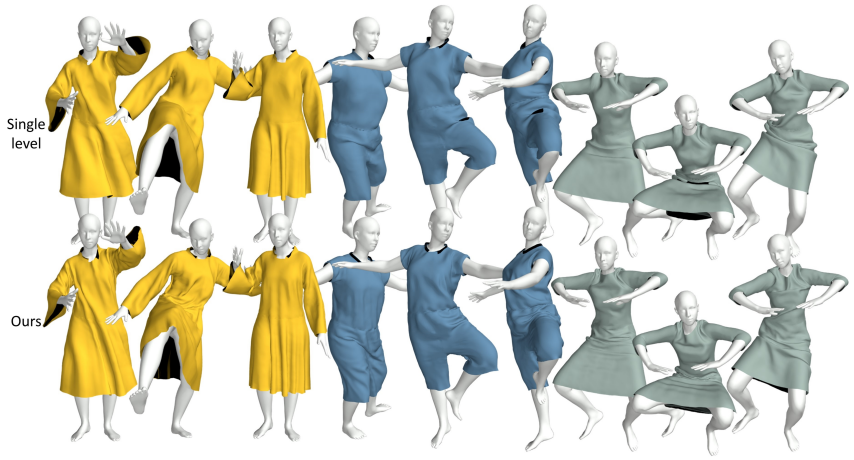

Fig. 4. The deformation results of the single-level network and ours.

network architecture. However, should the network lack sufficient depth, it could potentially fail to spread specific vertex information to the extent necessary for accurately simulating the information integral to clothing deformation. The pooling method provides a direct way to extend the influence range of a vertex. Within a network of equivalent depth, two pooling operations can increase the influence field of a vertex by four neighborhoods. To thoroughly evaluate the effects of multilevel processing, we construct a network devoid of pooling for comparison. In this setup, the GATs involved in multilevel processing are cascade connected to produce a single-level network.

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 has a substantial impact on the deformation results, these vertices will be highlighted in the heatmap generated by the attribution value calculations.

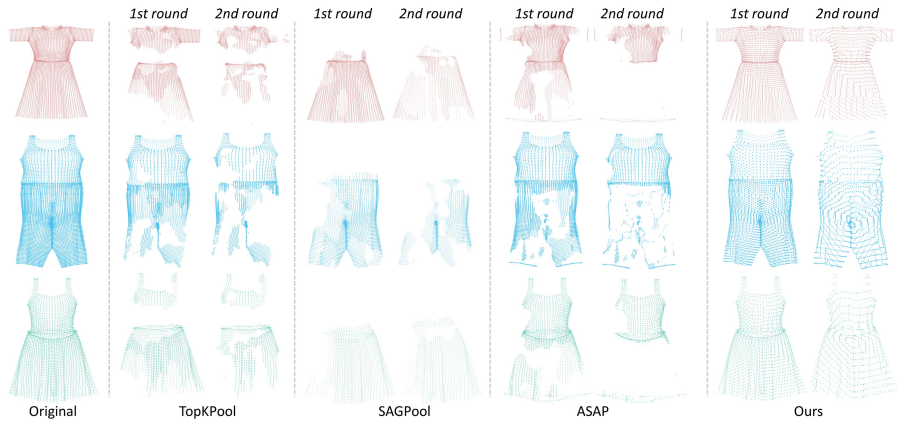

Fig. 5. Structure preservation between different graph pooling strategies. Starting from the original template dress, we show the results for pooling once and twice.

More specifically, we use an attribution calculation method [Shi et al. 2022] based on Aumann-Shapley values. 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:

$$R_i = X \sum_{k=1}^{N_s} \frac{1}{(1-\xi_k^2)\left[P'_{N_s}(\xi_k)\right]^2} \frac{\partial f_i\Big(1/2(1+\xi_k)X\Big)}{\partial X}, \tag{9}$$

where N_s is the number of sample points in Gauss-Legendre quadrature for approximating the definite integral. ξ_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 is generated using attribution result dimensions corresponding to normals.

The attribution results show the factual field influencing the vertex, assisting in the examination of specific deformation regions. As shown in Fig. 3, the vertices in our network receive a broader impact, resulting in a more comprehensive deformation. In the single-level 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. 4, the single-level deformation of the yellow dress does not create a coordination effect between the upper and lower body of the dress. Conversely, our deformation results tend to demonstrate superior consistency overall and can generate larger wrinkle areas on tested garments, as shown in the bottom line of Fig. 4. Moreover, our findings suggest that employing multilevel processing after two rounds of pooling and unpooling presents a beneficial balance between speed and quality. Quantitative results are also provided in first and last rows of Table 1. The experimental findings suggest that converting a cascade connected structure into a multilevel one is more conducive for simulating garment deformation.

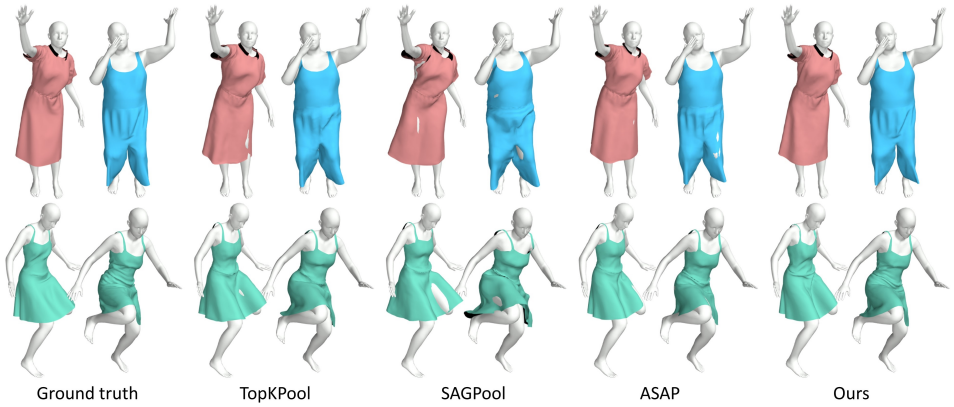

Fig. 6. Qualitative deformation results of applying different pooling strategies.

4.2 Evaluation on Different Pooling Strategies

We test the structural preservation capabilities of various pooling strategies on different meshes, specifically comparing ours with TopKPool [Gao and Ji 2022], SAGPool [Lee et al. 2019], and ASAP [Ranjan et al. 2019] strategies. As shown in Fig. 5, our approach effectively achieves mesh sparsification while preserving the global structure after one and two rounds of pooling. Unlike other methods, our approach pools all nodes on every alternate breadth-first-search frontier. This approach, both more direct and simple, ensures a comprehensive representation of the garment mesh. It guarantees that no specific region of the mesh is disproportionately overlooked, leading to a more uniform preservation of the mesh structure. 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 sometimes lead to the exclusion of vertices in specific regions, resulting in the loss of that structural region. Notably, ASAP is effective at maintaining boundaries of the mesh, but its preservation of the interior structure is less satisfactory. Overall, other methods might be more appropriate for graph data where node importance varies considerably, such as in protein or molecular graphs. In contrast, our method excels in preserving mesh structure, making it particularly adept for tasks involving garment deformation.

In addition, we present qualitative deformation results of applying different pooling strategies in Fig. 6. For unseen garments, body shapes, and motions, our proposed pooling method effectively captures global features related to deformations. This capability assists the model in generalizing to unseen test data and in producing reliable deformation predictions. Alternately, 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. Approaches exhibiting strong structure preservation tend to enhance the quality of the final deformation results. Quantitative results are also provided in Table 1. As the animation behavior of dresses is more complex, it is anticipated that dresses would have higher errors than jumpsuits. The results also reveal that not all employed graph pooling strategies consistently outperform the single-level method. In contrast, our method results in the lowest error, further demonstrating its suitability for the task of garment deformation.

Table 1. Quantitative deformation results of applying multilevel processing and different pooling strategies. For two types of garments, dresses and jumpsuits, we measure two metrics: the average vertex distance E_{verts} (mm) and the average angular deviation of vertex normals E_{norm} (°) between predictions and PBS.

	Jumpsuit E_{verts} / E_{norm}	Dress E_{verts} / E_{norm}
Single-level	29.27 (±45.62) / 17.72 (±18.37)	31.54 (±71.05) / 20.67 (±23.15)
TopKPool	29.95 (±47.27) / 18.57 (±19.58)	32.74 (±75.73) / 19.95 (±23.74)
SAGPool	32.04 (±53.03) / 19.35 (±20.85)	40.39 (±82.55) / 22.50 (±25.82)
ASAP	31.02 (±47.05) / 18.78 (±18.16)	32.92 (±74.37) / 20.35 (±21.89)
Ours	**22.94** (±27.61) / **12.75** (±13.80)	**24.93** (±48.56) / **16.61** (±21.72)

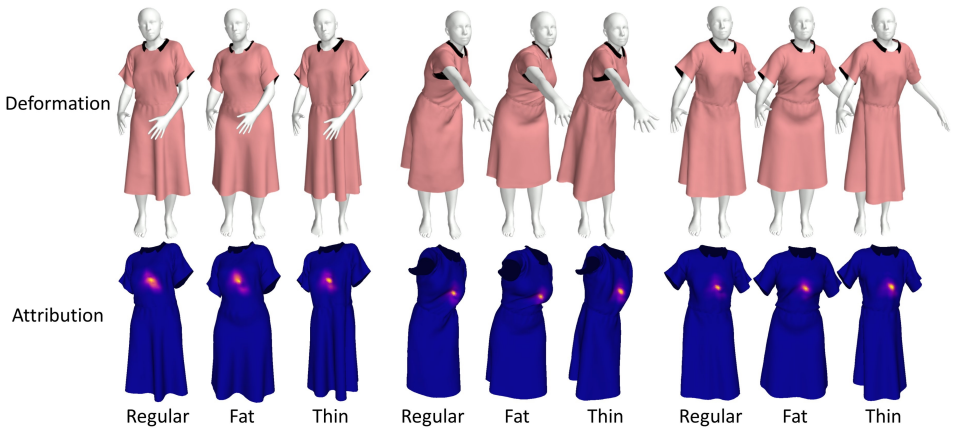

Fig. 7. The deformation and attribution results on three different body shapes.

4.3 Evaluation on Different Body Shapes

The experiments presented in the previous subsections affirm that our method is capable of producing natural deformation predictions for novel garments and motions. We further explore the impact of distinct body shapes (regular, fat, thin) on the deformation of the same (also unseen in training) garment. As depicted in Fig. 7, the first row exhibits the deformation results predicted by our method. It can be observed that the intricate folds in the garment are intimately related to the body shape, highlighting the effectiveness of our method in learning these types of deformation information. Moreover, we provide the corresponding attribution results in the second row. For a sample vertex, variation in the factual field influencing it across different body shapes is evident. This variation in the factual field confirms the capability of our method to adaptively extract information from neighboring vertices, while simultaneously exhibiting the adaptability of the model in capturing rich deformation patterns associated with different body shapes.

4.4 Comparison to Existing Methods

In our study, we benchmark our approach against three state-of-the-art methods, including DANet [Li et al. 2023a], NCS [Bertiche et al. 2022], and HOOD [Grigorev et al. 2023]. Their primary distinctions lie in their ability to generate realistic deformations on diverse garments, and their speed of inference, as depicted in Table 2. A check mark in the table indicates full capability, whereas an X mark signifies a lack of ability. DANet lacks dynamic processing capabilities, it looks very

Table 2. Comparison with state-of-the-art methods in garment generalization, batch support, and inference speed.

	Dynamics	Generalization	Batch Support	Speed
DANet	✗	✓	✓	297.3
NCS	✓	✗	✓	853.9
HOOD	✓	✓	✗	13.4
Ours	✓	✓	✓	402.1

unnatural when processing sequence actions. Although NCS show robust generalization to novel motions, they fall short of processing diverse garments within a unified network. Contrastingly, our method can generate detailed and realistic deformations for diverse garments at a hard real-time speed. This increases the practical appeal of our method, particularly in scenarios requiring multiple garment types. For more replication details, please refer to the supplemental materials.

For inference time, we observed a significant increase of more than 30% in runtime during the replication of NCS. This occurred even though we retain identical network structures, and the same input and output configurations as the original NCS implementation. This discrepancy might be attributed to the backend architecture of the deep learning frameworks. Hence, we directly report the time performance results as presented in the original paper. DANet, NCS, and our method all deliver results that can also be categorized as hard real-time. Both DANet and our method leverage graph learning for garment mesh deformation; however, they rely on a two-step process for predicting deformations. In contrast, as demonstrated in Sec. 3.2, our model is compact and streamlined, integrating specialized components for handling different types of features. It utilizes temporal weights and blend shapes to directly compensate for deformation details, enhancing the efficiency of deformation estimation. On the other hand, HOOD fails to achieve this level of efficiency as it cannot process motion sequences in batches, diminishing the value of using deep neural networks. Another important difference is that NCS and HOOD are trained with unsupervised losses. Despite their advantage in reducing data preparation compared to our supervised method, their lack of effective evaluation indicators during training makes it challenging to control the overall training progress. In contrast, supervised losses allow us to directly monitor the optimization status. Moreover, the test results in Fig. 8 demonstrate that our method achieves the closest approximation to the ground truth.

The ultimate aim of garment deformation is to enhance the user experience. To assess this, we create animation videos using three garments for a user study, with some qualitative results shown in Fig. 8. We recruited 35 participants for this study, 12 with expertise in graphics and 23 with computer science knowledge but no specific knowledge of computer graphics. Participants were asked to watch all animations, presented in a random order, and subsequently rate each on the basis of realism and detail richness separately. The rating system comprised four levels, ranging from 0 to 3, with each score being assigned only once. Participants were allowed to replay any animation multiple times during the scoring process. As shown in Fig. 9, our method generally outperforms others in terms of detail richness, particularly in creating detailed wrinkles in the waist area of the short jumpsuit. On the other hand, we discover NCS exhibits superior performance in terms of realism. However, it is important to highlight that the NCS model was trained specifically with these garments, a condition not replicated with the other methods during training.

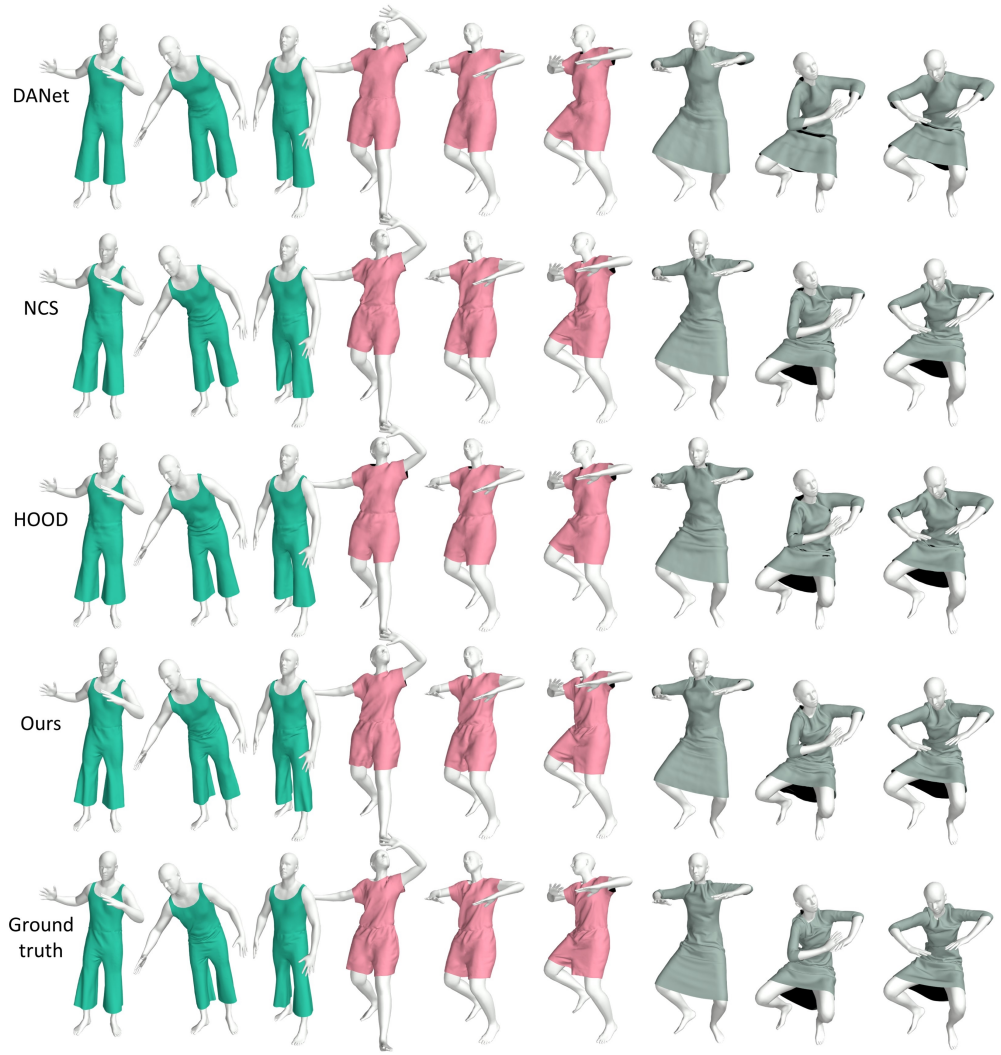

Fig. 8. Examples from different methods used in the user study.

5 CONCLUSIONS

We have presented a graph learning-based method for predicting fine-level deformations of complicated garment deformation, achieving a speed approximately 30 times faster than physics-based simulators. Our unified model demonstrates the ability to generalize to unseen garments, body shapes, and poses, without compromising deformation quality. To realize this, we initially propose an innovative, comprehensive representation of the garment graph which efficiently incorporates deformation cues from multiple information sources. Subsequently, we introduce utilitarian pooling and unpooling strategies for maintaining graph structure, and design a multilevel processing module to ensure effective feature transition within the graph. The efficient representation and processing of features contribute to the generation of realistic garment deformations. Lastly, we

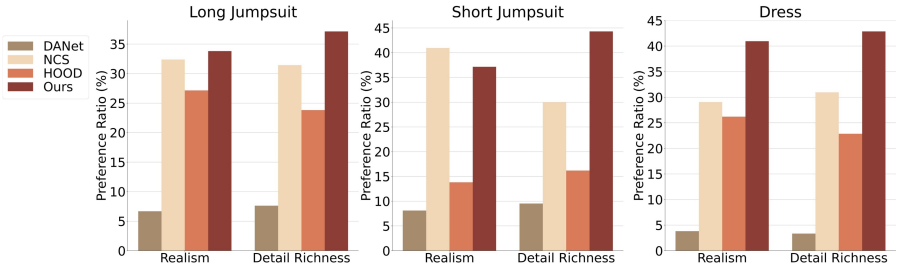

Fig. 9. User evaluations of the realism and detail richness in the garment deformations generated by the tested methods.

utilize attribution analysis to enhance the interpretability of our approach, which show practical ability in case analysis.

There are also some limitations that could be addressed by follow-up works. Despite our method showing superior performance on self-collision in comparison to self-supervised methods, fully resolving self-collisions, particularly in garments with complicated folds and overlaps, remains a complex challenge. The design of appropriate loss functions or the development of innovative post-processing methods could be promising direction for future exploration. 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.

ACKNOWLEDGEMENT

This work has been partially supported by Beijing Natural Science Foundation (No. 4244088, 4232017) and JSPS KAKENHI (No. 22K12331).

REFERENCES

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2020. CLOTH3D: clothed 3d humans. In *Proc. Eur. Conf. Comput. Vis.* 344–359.

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2021. PBNS: Physically Based Neural Simulation for Unsupervised Garment Pose Space Deformation. *ACM Trans. Graph.* 40, 6, Article 198 (2021). https://doi.org/10.1145/3478513.3480479

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2022. Neural Cloth Simulation. *ACM Trans. Graph.* 41, 6, Article 220 (2022). https://doi.org/10.1145/3550454.3555491

Carnegie-Mellon. 2010. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/. Accessed: 2023.

Zhen Chen, Hsiao-Yu Chen, Danny M. Kaufman, Mélina Skouras, and Etienne Vouga. 2021. Fine Wrinkling on Coarsely Meshed Thin Shells. *ACM Trans. Graph.* 40, 5, Article 190 (2021). https://doi.org/10.1145/3462758

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph.* 21, 3 (jul 2002), 604–611. https://doi.org/10.1145/566654.566624

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014a. Yarn-Level Simulation of Woven Cloth. *ACM Trans. Graph.* 33, 6, Article 207 (2014). https://doi.org/10.1145/2661229.2661279

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014b. Yarn-Level Simulation of Woven Cloth. *ACM Trans. Graph.* 33, 6, Article 207 (2014). https://doi.org/10.1145/2661229.2661279

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. 2004. Spectral grouping using the Nystrom method. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 2 (2004), 214–225. https://doi.org/10.1109/TPAMI.2004.1262185

Hongyang Gao and Shuiwang Ji. 2022. Graph U-Nets. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 9 (2022), 4948–4960. https://doi.org/10.1109/TPAMI.2021.3081010

Artur Grigorev, Bernhard Thomaszewski, Michael J Black, and Otmar Hilliges. 2023. HOOD: Hierarchical Graphs for Generalized Modelling of Clothing Dynamics. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 16965–16974. https://doi.org/10.1109/CVPR52729.2023.01627

Erhan Gundogdu, Victor Constantin, Shaifali Parashar, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua. 2022. GarNet++: Improving Fast and Accurate Static 3D Cloth Draping by Curvature Loss. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 1 (2022), 181–195. https://doi.org/10.1109/TPAMI.2020.3010886

Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, Robert W. Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross. 2014. Subspace Clothing Simulation Using Adaptive Bases. *ACM Trans. Graph.* 33, 4, Article 105 (2014). https://doi.org/10.1145/2601097.2601160

Moon-Hwan Jeong, Dong-Hoon Han, and Hyeong-Seok Ko. 2015. Garment capture from a photograph. *Comput. Animat. Virtual Worlds* 26 (2015), 291–300. https://doi.org/10.1002/cav.1653

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

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-Attention Graph Pooling. In *Proc. Int. Conf. Mach. Learn.*, Vol. 97. 3734–3743.

John P. Lewis, Cordner Matt, and Fong. Nickson. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proc. Annu. Conf. Comput. Graph. Interact. Tech.* 165–172. https://doi.org/10.1145/344779.344862

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E. Brown, and Laurence Boissieux. 2018. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph.* 37, 4, Article 52 (2018). https://doi.org/10.1145/3197517.3201308

Tianxing Li, Rui Shi, and Takashi Kanai. 2020. DenseGATs: A Graph-Attention-Based Network for Nonlinear Character Deformation. In *Proc. Symp. Interactive 3D Graph. Games.* 5:1–5:9. https://doi.org/10.1145/3384382.3384525

Tianxing Li, Rui Shi, and Takashi Kanai. 2021. MultiResGNet: Approximating Nonlinear Deformation via Multi-Resolution Graphs. *Comput. Graph. Forum* 40, 2 (2021), 537–548. https://doi.org/10.1111/cgf.142653

Tianxing Li, Rui Shi, and Takashi Kanai. 2023a. Detail-Aware Deep Clothing Animations Infused with Multi-Source Attributes. *Comput. Graph. Forum* 42, 1 (2023), 231–244. https://doi.org/10.1111/cgf.14651

Tianxing Li, Rui Shi, Qing Zhu, and Takashi Kanai. 2023b. SwinGar: Spectrum-Inspired Neural Dynamic Deformation for Free-Swinging Garments. *IEEE Trans. Vis. Comput. Graphics* (2023), 1–16. https://doi.org/10.1109/TVCG.2023.3346055

Mario Lino, Chris D. Cantwell, Anil A. Bharath, and Stathi Fotiadis. 2021. Simulating Continuum Mechanics with Multi-Scale Graph Neural Networks. *arXiv:2106.04900* (2021).

Mario Lino, Stathi Fotiadis, Anil A. Bharath, and Chris D. Cantwell. 2022. Multi-scale rotation-equivariant graph neural networks for unsteady Eulerian fluid dynamics. *Phys. Fluids* 34, 8 (08 2022), 087110. https://doi.org/10.1063/5.0097679

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 (2015). https://doi.org/10.1145/2816795.2818013

N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. 1989. Joint-Dependent Local Deformations for Hand Animation and Object Grasping. In *Proc. Graph. Interface.* 26–33.

Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph.* 31, 6, Article 152 (2012). https://doi.org/10.1145/2366145.2366171

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

Xiaoyu Pan, Jiaming Mai, Xinwei Jiang, Dongxue Tang, Jingxiang Li, Tianjia Shao, Kun Zhou, Xiaogang Jin, and Dinesh Manocha. 2022. Predicting Loose-Fitting Garment Deformations Using Bone-Driven Motion Networks. In *ACM SIGGRAPH.* Article 11. https://doi.org/10.1145/3528233.3530709

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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 7363–7373. https://doi.org/10.1109/CVPR42600.2020.00739

Ekagra Ranjan, Soumya Sanyal, and Partha Pratim Talukdar. 2019. ASAP: Adaptive Structure Aware Pooling for Learning Hierarchical Graph Representations. In *Proc. AAAI Conf. Artif. Intell.* https://api.semanticscholar.org/CorpusID:208158156

Igor Santesteban, Miguel Otaduy, Nils Thuerey, and Dan Casas. 2022b. ULNeF: Untangled Layered Neural Fields for Mix-and-Match Virtual Try-On. In *Proc. Adv. Neural Inf. Process. Syst.*, Vol. 35. 12110–12125.

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

Igor Santesteban, Miguel A. Otaduy, and Dan Casas. 2022a. SNUG: Self-Supervised Neural Dynamic Garments. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 8130–8140. https://doi.org/10.1109/CVPR52688.2022.00797

Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. 2021. Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 11758–11768. https://doi.org/10.1109/CVPR46437.2021.01159

Rui Shi, Tianxing Li, and Yasushi Yamaguchi. 2022. Output-targeted baseline for neuron attribution calculation. *Image Vis. Comput.* 124 (2022), 104516. https://doi.org/10.1016/j.imavis.2022.104516

Leonid Sigal, Moshe Mahler, Spencer Diaz, Kyna McIntosh, Elizabeth Carter, Timothy Richards, and Jessica Hodgins. 2015. A Perceptual Control Space for Garment Simulation. *ACM Trans. Graph.* 34, 4, Article 117 (2015). https://doi.org/10.1145/2766971

Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. 2010. Video-Based Reconstruction of Animatable Human Characters. In *ACM SIGGRAPH Asia*. Article 139. https://doi.org/10.1145/1866158.1866161

Min Tang, Huamin Wang, Le Tang, Ruofeng Tong, and Dinesh Manocha. 2016. CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation. *Comput. Graph. Forum* 35, 2 (2016), 511–521. https://doi.org/10.1111/cgf.12851

Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. 2020. SIZER: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing. In *Proc. Eur. Conf. Comput. Vis.*, Vol. 12348. 1–18. https://doi.org/10.1007/978-3-030-58580-8_1

Tzvetomir Vassilev, Spanlang Bernhard, and Chrysanthou. Yiorgos. 2001. Fast Cloth Animation on Walking Avatars. *Computer Graphics Forum* 20, 3 (2001), 260–267. https://doi.org/10.1111/1467-8659.00518

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proc. Int. Conf. Learn. Representations*. https://openreview.net/forum?id=rJXMpikCZ

Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. 2020. Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *Comput. Graph. Forum* 39, 8 (2020), 145–156. https://doi.org/10.1111/cgf.14109

Huamin Wang. 2021. GPU-Based Simulation of Cloth Wrinkles at Submillimeter Levels. *ACM Trans. Graph.* 40, 4, Article 169 (2021). https://doi.org/10.1145/3450626.3459787

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 (2018). https://doi.org/10.1145/3272127.3275074

Longhua Wu, Botao Wu, Yin Yang, and Huamin Wang. 2020. A Safe and Fast Repulsion Method for GPU-Based Cloth Self Collisions. *ACM Trans. Graph.* 40, 1, Article 5 (2020). https://doi.org/10.1145/3430025

Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C. Lin. 2018. Physics-Inspired Garment Recovery from a Single-View Image. *ACM Trans. Graph.* 37, 5, Article 170 (2018). https://doi.org/10.1145/3026479

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Proc. Adv. Neural Inf. Process. Syst.* 4805–4815.

A APPENDIX

Algorithm 1: Pooling a fine graph $\mathcal{G}^0$ to generate a coarse graph $\mathcal{G}^c$.

input : original graph $\mathcal{G}^0 = (\mathcal{X}^0, \mathcal{E}^0)$, first-order neighboring $\mathcal{N}^0$
output: coarse graph $\mathcal{G}^c = (\mathcal{X}^c, \mathcal{E}^c)$

$e^{ecc} \leftarrow$ eccentricity of $\mathcal{G}^0$;
$x_{center} \leftarrow$ select first from $\{ x_i \in \mathcal{X}^0 | e_i^{ecc} = min(e^{ecc}) \}$;
for x_i *in* $\mathcal{X}^0$ **do**
  $d_i \leftarrow$ geodesic distance of (x_{center}, x_i) ;
end
$\mathcal{X}^c \leftarrow \{x_i \in \mathcal{X}^0 \mid d_i \mod 2 = 0\}$;
$\mathcal{X}^r \leftarrow \{x_i \in \mathcal{X}^0 \mid d_i \mod 2 = 1\}$;
for x_i *in* $\mathcal{X}^r$ **do**
  $\mathcal{X}^{from} \leftarrow \{x_j \in \mathcal{X}^0 \mid e_{ij} \in \mathcal{E}^0 \ \& \ d_j = d_i - 1\}$;
  $\mathcal{X}^{to} \leftarrow \{x_j \in \mathcal{X}^0 \mid e_{ij} \in \mathcal{E}^0 \ \& \ d_j = d_i + 1\}$;
  for x_j *in* $\mathcal{X}^{from}$ **do**
    for x_k *in* $\mathcal{X}^{to}$ **do**
      $add(e_{jk}, \mathcal{E}^c)$
    end
  end
end
for x_i *in* $\mathcal{X}^c$ **do**
  for j *in* $\mathcal{N}_i^0$ **do**
    if $x_j \in \mathcal{X}^c \ \& \ e_{ij} \notin \mathcal{E}^c$ **then**
      $add(e_{ij}, \mathcal{E}^c)$
    end
  end
end

Algorithm 2: Unpooling a coarse graph $\mathcal{G}^c$ to generate a fine graph $\mathcal{G}^0$.

input : coarse graph $\mathcal{G}^c = (\mathcal{X}^c, \mathcal{E}^c)$, fine graph edges $\mathcal{E}^0$, removed vertex features $\mathcal{X}^r$, first-order neighboring $\mathcal{N}^0$
output: fine graph $\mathcal{G}^0 = (\mathcal{X}^0, \mathcal{E}^0)$

Initialize elements x_i^0 in $\mathcal{X}^0$ = with zero ;
$\mathcal{N}_i^{c0} \leftarrow \{j | j \in \mathcal{N}_i^0 \ \& \ x_j \in \mathcal{X}^c\}$;
$\mathcal{N}_i^{r0} \leftarrow \{j | j \in \mathcal{N}_i^0 \ \& \ x_j \in \mathcal{X}^r\}$;
for x_i^r *in* $\mathcal{X}^r$ **do**
  for j *in* $\mathcal{N}_i^{c0}$ **do**
    $\hat{w}_{ij} = \frac{A_{ij}}{\sum_{i' \in \mathcal{N}_j^{r0}} A_{i'j}}$;
  end
end
for x_i^c *in* $\mathcal{X}^c$ **do**
  $x_i^0 = x_i^c$;
end
for x_i^r *in* $\mathcal{X}^r$ **do**
  $w_{ij} = \frac{\hat{w}_{ij}}{\sum_{j \in \mathcal{N}_i^{c0}} \hat{w}_{ij}}$;
  $x_i^0 = \sum_{j \in \mathcal{N}_i^{c0}} w_{ij} x_j^c$;
end

Supplemental Materials of Efficient Deformation Learning of Varied Garments with a Structure-Preserving Multilevel Framework

1 NETWORK ARCHITECTURE

Our method can be divided into two stages: 1) garment-body information encoding stage, and 2) deformation generation stage. Although both stages necessitate computation for parameter optimization during training, the initial encoding stage requires calculation only once for predicting deformation.

The initial stage of our method separately processes body shape, garment graph information, and garment type information. The body shape encoder ($\mathcal{E}^{bs}$) processes the distances of body vertices to joints through three linear layers with ReLU activation, containing [512, 256, 128] channels respectively. The garment graph encoder ($\mathcal{E}^{gg}$) manages the garment graph in its rest pose through three GAT blocks with ReLU activation, each containing [32, 64, 128, 128] channels and 24 heads. For this, we form the garment graph using garment mesh vertices, normals, and distances to the body joints, resulting in 30 feature channels in the input graph. The last GAT block here is used to process the multiplication of the garment style cues and garment graph features. The garment style encoder ($\mathcal{E}^{gs}$) processes the eigenvalues of the approximated affinity matrix into a 128-length vector with three linear layers with ReLU activation, each containing [256, 256, 128] channels. The garment-body encoding is subsequently computed via multiplication of the outputs from these three encoders.

The second stage, which generates deformations, comprises two temporal encoders, $\mathcal{E}^{t1}$ and $\mathcal{E}^{t2}$, along with a multi-level processing module, $\mathcal{M}^{mp}$. The temporal encoder $\mathcal{E}^{t1}$, which handles pose sequences including rotation angles and the body's translation relative to the previous frame, is equipped with a two-layer LSTM with 128 channels. The skinning weights are generated by multiplying the garment-body encoding with the temporal features from $\mathcal{E}^{t1}$, and softmax is applied to ensure the sum of all joint weights equals 1. The temporal encoder $\mathcal{E}^{t2}$, which processes the flattened body vertex velocities, consists of a two-layer LSTM with 128×24 channels. The input for the multi-level processing module is the element-wise multiplication of the garment-body encoding and the temporal features from $\mathcal{E}^{t2}$. $\mathcal{M}^{mp}$ comprises a pair of submodules with identical structures, one optimized by supervised loss and the other by unsupervised loss. For each submodule, we first apply a GAT to embed motion-informed features into a fixed dimension. After the graph embedding, we build the encoder consisting of several encoder blocks, each of which contains a pooling layer and a GAT block. The pooling layers reduce the size of the fine graph, while GAT blocks are responsible for aggregating the first-order information of the coarse graph. Then, we build the decoder part where contains the same number of decoder blocks. Each decoder block is composed of an unpooling layer and a GAT block. There are additive skip-connections between corresponding blocks of encoder and decoder layers, which transmit graph structure information for better performance. Finally, a GAT block is employed to generate blend shape results. In each submodule, barring the final output GAT block, each GAT block contains 32 channels and 4 heads with tanh activation. The final GAT block has 3 channels and 4 heads, using an averaging approach for head processing to generate the blend shape.

2 LOSS FUNCTIONS

Our loss terms and hyperparameter balance strategy are borrowed from several previous studies [Bertiche et al. 2022, 2021; Santesteban et al. 2022] with slight modification. Here, we describe these loss terms in detail.

Supervised loss. We use vertex position, edge length, and norm angle as our supervised loss terms.

$$\mathcal{L}_{\text{vert}} = k_v \frac{1}{N} \sum_i^N \|x_i - x_i^{\text{GT}}\|_2, \tag{1}$$

$$\mathcal{L}_{\text{edge}} = k_e \|E - E^{\text{GT}}\|_2^2, \tag{2}$$

$$\mathcal{L}_{\text{norm}} = k_n \frac{1}{N} \sum_i^N (\mathbf{1} - n_i \cdot n_i^{\text{GT}}), \tag{3}$$

where the variable with superscript "GT" means the ground truth data. x_i is the vertex position, E is the edge length, and n_i is the normal vector. The hyperparameters k_v, k_e, and k_n are set to 100, 15, and 50, respectively.

Unsupervised loss. We use collision, gravity, stretch force, and shearing force as our unsupervised loss terms.

$$\mathcal{L}_{\text{collision}} = k_c \sum_i^N \left|\min\left(h(x_i) - \epsilon, 0\right)\right|, \tag{4}$$

$$\mathcal{L}_{\text{gravity}} = -k_g \sum_i^N m_i x_i \cdot g, \tag{5}$$

$$\mathcal{L}_{\text{stretch}} = k_s \sum_i^{N_F} a_i \left(\|W_i^u\|_2 - 1\right)^2 + a_i (\|W_i^v\|_2 - 1)^2, \tag{6}$$

$$\mathcal{L}_{\text{shear}} = k_h \sum_i^{N_F} a_i \left(W_i^u \cdot W_i^v\right)^2, \tag{7}$$

$$(W^u W^v) = \begin{pmatrix} \Delta x_1 & \Delta x_2 \end{pmatrix} \begin{pmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{pmatrix}^{-1}, \tag{8}$$

where $h(x)$ is the signed distance to a body mesh and ϵ is a collision threshold value. 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. $\Delta x_1 = x_j - x_i$ and $\Delta x_2 = x_k - x_i$. The indices i, j, k are face triangle vertices. The vertex position x_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 Δx. The hyperparameters k_c, k_g, k_s, and k_h are set to 1.5, 0.1, 5, and 2, respectively.

3 DATASET GENERATION

Regarding garment variation, as depicted in Figure 1, we collect 55 different garments, which include a variety of dresses and jumpsuits from the publicly accessible CLOTH3D dataset [Bertiche et al. 2020] for the training process. It is noteworthy mention that in the experimental section of the main text, we present results for test garments, all of which are entirely novel and not included in the training data.

Regarding body shape variation, we follow a similar selection principle as outlined in [Patel et al. 2020], where the first two components of β are systematically sampled at four evenly spaced points

Fig. 1. Garment samples used for training.

within the interval $[-2, 2]$, while keeping all other components constant. Through this process, eight distinct body shapes are produced, complemented by a canonical zero shape to constitute the complete set of nine training samples for body data. In the testing phase, we continue randomizing these first two components across the range of $[-2, 2]$, ensuring that no test samples overlap with those used during training.

4 NETWORK TRAINING

Inspired by the unsupervised learning strategy for neural simulation of garment deformation [Bertiche et al. 2022, 2021], we also design a two-step training strategy that partitions network training into two sequential steps. The first step optimizes network parameters with supervised loss, including vertex position, vertex normal, and edge length. We adjust different loss hyperparameters based on the principle that loss terms should have comparable magnitudes. This initial training is critical for producing stable garment-body encoding and skinning weights. When the vertex position error reduction rate over 50 epochs is less than 4%, we freeze most parameters and only update the parameters of the second submodule of $\mathcal{M}^{mp}$ in subsequent training. In the second step, we scale down the vertex loss and edge length loss by a factor of 0.1, close the normal loss, then activate collision and gravity loss. When the collision term reduction rate over 20 epochs is less than 5%, we start to activate all other unsupervised losses, including stretch and shear losses. The training batch size is set to 64. We employ the Adamax optimizer with an initial learning rate of 3e-3 and use cosine annealing for learning rate decay. Default Glorot initialization is used for GAT blocks, while the Kaiming method is employed for LSTM and linear layers. Network training is conducted on two servers, each equipped with two nVIDIA RTX A6000 GPUs and two H800 GPUs, respectively. We evaluate the results on a computer fitted with an I9 13900K CPU and an nVIDIA GeForce RTX 4090 GPU, using a batch size of 16.

5 REPLICATION DETAILS

For DANet, in addition to the vertex loss, we incorporate edge and norm losses. This addition can expedite the convergence during training, particularly when handling larger datasets. For the NCS implementation, we slightly alter the strategy for generating skinning weights, an essential factor in creating realistic long-dress deformations. Specifically, we sample body skinning weights around a particular garment vertex and average them to produce the final weights. Moreover, the inertial loss value shows a complex variation pattern based on garment types and motions. Empirically, for long dresses, it is beneficial to initially freeze the inertial loss and then progressively activate it once the collision loss smoothens. Starting from 1/10 and gradually increasing to 1 unit, this processing enhances convergence stability and yields more realistic results. For HOOD, we find the challenge in replication lies in the initial pose complexity. When dealing with a complex initial pose, it becomes challenging to achieve a garment with better internal energy, a crucial aspect of HOOD initialization. Despite being an unsupervised method, HOOD requires more extensive data preparation efforts compared to NCS. For unsupervised studies, the Saint Venant Kirchhoff elastic material model is used as the unsupervised cloth model loss due to its demonstrated effectiveness in [Bertiche et al. 2022; Grigorev et al. 2023; Santesteban et al. 2022].

REFERENCES

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2020. CLOTH3D: clothed 3d humans. In *Proc. Eur. Conf. Comput. Vis.* 344–359.

Hugo Bertiche, Meysam Madadi, and Sergio Escalera. 2022. Neural Cloth Simulation. *ACM Trans. Graph.* 41, 6, Article 220 (2022). https://doi.org/10.1145/3550454.3555491

Hugo Bertiche, Meysam Madadi, Emilio Tylson, and Sergio Escalera. 2021. DeePSD: Automatic deep skinning and pose space deformation for 3D garment animation. In *Proc. IEEE Int. Conf. Comput. Vis.* 5471–5480.

Artur Grigorev, Bernhard Thomaszewski, Michael J Black, and Otmar Hilliges. 2023. HOOD: Hierarchical Graphs for Generalized Modelling of Clothing Dynamics. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 16965–16974. https://doi.org/10.1109/CVPR52729.2023.01627

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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 7363–7373. https://doi.org/10.1109/CVPR42600.2020.00739

Igor Santesteban, Miguel A. Otaduy, and Dan Casas. 2022. SNUG: Self-Supervised Neural Dynamic Garments. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 8130–8140. https://doi.org/10.1109/CVPR52688.2022.00797