Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Research/Review Article

GarTrans: Transformer-Based Architecture for Dynamic and Detailed Garment Deformation

Tianxing Li[1], Zhi Qiao[2](✉), Zihui Li[1], Rui Shi[3,4], and Qing Zhu[1]

Abstract In this paper, we introduce GarTrans, a novel graph-learning based method for the task of garment animation. It emphasizes efficiently rendering realistic deformation effects. GarTrans goes beyond existing models by providing improved generalization capabilities, along with the ability to capture fine-scale garment dynamics and details. Our approach begins by constructing a garment graph that comprehensively encodes the dynamic state of the garment, taking into account its shape and topology, as well as the underlying body shape and corresponding motion. We have also designed a structure-augmented transformer (SAT) capable of processing the node information and edges within the graph, enabling the generation of deformation details that are contextually informed. Our model employs a unified optimization scheme that incorporates both supervised and unsupervised loss functions, enabling a robust approach capable of realistically mimicking the behavior of intricate garments. Experimental evaluations show that our method surpasses the existing state-of-the-art in terms of both functional capabilities and visual fidelity, advancing the field of garment animation.

Keywords cloth deformation, dynamics, generalization, graph learning

1 College of Computer Science, Beijing University of Technology, Beijing, 100124, China. E-mail: T. Li, litianxing@bjut.edu.cn; Z. Li, lizihui@emails.bjut.edu.cn; Q. Zhu, ccgszq@bjut.edu.cn.

2 College of Information and Electrical Engineering, China Agriculture University, Beijing, 100083, China. E-mail: zhiqiao223@cau.edu.cn.

3 School of Information Science and Technology, Beijing University of Technology, Beijing, 100124, China. E-mail: R. Shi, ruishi@bjut.edu.cn.

4 Department of General Systems Studies, the University of Tokyo, Tokyo, 153-8902, Japan.

Manuscript received: 2023-12-22; accepted: 2024-06-18

1 Introduction

Efficient and realistic simulation of garments is indispensable to human modeling in industries such as gaming, film, e-commerce, and digital twins. In the field of computer graphics, the classic approach to modeling clothing behavior is physics-based simulation (PBS) [1–3]. While it generates high-quality results, the process is not only challenging to manage, requiring the specialist expertise of an animator, but it also incurs significantly high computational costs, making it unsuitable for interactive applications. Seeking efficiency in simulation, alternative techniques such as linear blend skinning (LBS) [4] or pose space deformation [5] are often employed. These methods operate under the assumption that the garment will closely follow body movement. While this speeds up the process, it does so at the cost of deformation realism.

Faced with the challenge of balancing simulation efficiency and quality, researchers are exploring learning-based solutions [6–8]. A prevalent approach among existing methods involves leveraging network structures such as multilayer perceptrons (MLPs) [9, 10] to map static garment deformations from input features like shape, pose, and clothing style. When dealing with loose-fitting garments, and aiming to produce dynamic effects, enhancements like gated recurrent units (GRUs) have been integrated to handle temporal actions [11, 12]. Nonetheless, these techniques often hit a common roadblock: they have difficulties generalizing to unseen garments and mesh topologies. While some unsupervised methods [13, 14] seem promising in minimizing data training time, their effectiveness in practical applications, particularly in scenarios encompassing a wide variety of garments, remains to be conclusively proven. The need for repeated retraining of models to cater to various garment types underscores the inefficiency of existing solutions. Even the latest research [15] efforts to enhance model generalizability using hierarchical graphs struggle with efficiency due to the

complexity of parallel sequence processing, resulting in an underutilization of GPU parallel resources.

The field urgently needs an efficient, comprehensive model capable of adeptly generating highly-detailed, physically accurate deformations for a diverse range of garments, inherently accounting for dynamic states of movements. To this end, we propose a novel approach, *GarTrans*, for predicting garment deformations using a transformer-based architecture. This approach is designed to understand the behavior of garments in line with the laws of physics. Furthermore, it guarantees that the resulting deformations are controllable and can be accurately guided by examples within the seen data. Fig. 1 illustrates the two-step pipeline of our method. Initially, we formulate a dynamic-aware graph construction module. This module generates garment graphs that implicitly encode information about individual garments, the underlying body, and motion states. Subsequently, we leverage this dynamic graph representation in a dynamic detail deformation module. This module is equipped with several graph transformers, which are responsible for producing the final garment deformation.

Specifically, our technical contributions are as follows:

- To address the challenge of the model's generalization capabilities under the variation of multiple deformation factors, we introduce a dynamic-aware graph construction module. This module constructs a comprehensive and unified representation that captures the inherent properties of the garment, the shape of the body, and the associated motion states. Through the integration of pose histories within the graph structure, our approach achieves temporal coherence of deformations, thereby ensuring realistic dynamic behavior of the garment.
- To gain a sophisticated understanding of how garment details behave in response to changes in body movement, we extend the graph transformer by proposing a dedicated pathway for mesh structural information. By allowing edge features to participate in the propagation, our approach maintains a 'live' state of the spatial relationships. This enables mesh nodes to more effectively perceive contextual details and facilitates dynamic interactions between information within the mesh. For garment deformation tasks, such structural augmentation contributes to the accurate prediction of details.
- To refine the fidelity of garment deformation predictions, we propose a novel optimization scheme that combines supervised and unsupervised loss functions. This hybrid approach harnesses the precision of supervised learning to meticulously guide the deformation process. Concurrently, it employs unsupervised loss functions, which encompass key physical qualities, to promote realistic garment behavior in simulations. This duality ensures that our network is not merely imitating specific instances but is learning the underlying physics and fabric characteristics, resulting in a robust model that excels in both precision and realism for the intricate garment deformation task.

Overall, our method provides a holistic solution capable of learning about garments with a reasonable sense of dynamism and detail using a unified optimization strategy.

Extensive experiments confirm the efficacy of our technologies and demonstrate its superiority to state-of-the-art learning-based approaches.

2 Related work

Existing work on garment deformation can be divided into two main categories: physics-based simulations and learning-based models.

Physics-based simulation (PBS) relies on the principles of physics to authentically reproduce the behavior and deformation of fabric under various forces and conditions. This technique is based on fundamental concepts such as mass, elasticity, gravity, and friction to create dynamic and realistic garment simulations. Numerous research avenues, including time integration [16], differentiable physics simulators [1, 17, 18], collision detection [19–21], and response [22–24], have been explored in this field. Despite the remarkable realism these methods provide, their substantial computational demands often limit their usability in interactive settings. To enhance simulation efficiency, several studies have focused on various facets, including incorporating position-based dynamics [25–27], leveraging parallel computation using GPUs [28–30], and refining low-resolution simulations by adding detailed wrinkles [31–33]. Another challenge in PBS is the tuning of simulation parameters, a process that demands both time and professional expertise. While researchers have introduced parameter inference methods to address this issue, they have yet to overcome the challenge of extending their effectiveness beyond controlled settings and ensuring consistent simulation outcomes across a variety of fabric types and dynamic conditions, which limits their practical applicability to diverse real-world scenarios.

Learning-based models have gained popularity for their ability to leverage existing machine learning techniques to learn and predict complicated garment behaviors, often

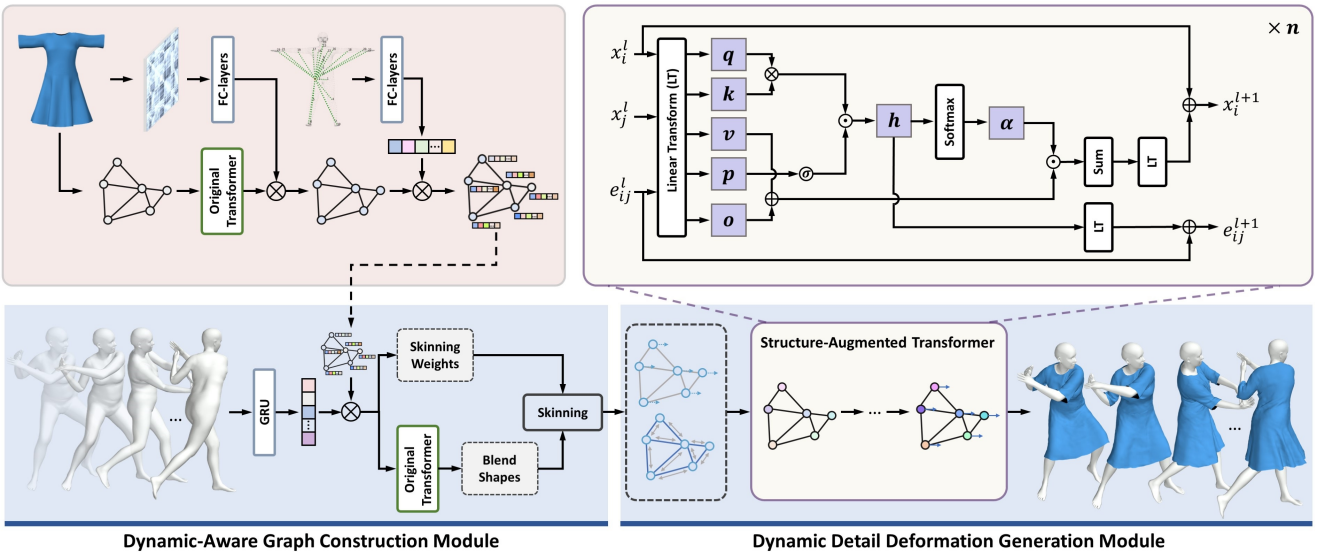


Fig. 1 Overview of our method. Inputs to the model include the garment and body in rest pose, alongside the continuous motion. GarTrans employs a dynamic graph construction module and a subsequent dynamic detail deformation generation module to process these inputs end-to-end, resulting in the output of final garment deformations that encompass dynamic behavior and rich details.

achieving higher efficiency and scalability than traditional physics-based approaches. Drawing on the foundational pose space deformation method [5], which is primarily used to simulate the deformation of animated characters, researchers have established mappings from variables such as body shape [34], garment size [35], fit parameters [36], or pose [10, 37], to the corresponding garment deformations. These mappings allow for automatic simulation of how garments react to different factors. To achieve realistic garment rendering with details, recent studies [38, 39] have adopted transformer-based networks for the task of garment deformation estimation. Despite their advances, these methods are restricted to garments with unchanging topologies. On the other hand, approaches in [38, 40] require the generation of a coarse mesh, which increases the time and computational effort needed. More recently, drawing inspiration from physics-based deformation models, authors have shifted their approaches towards unsupervised learning [41, 42], transforming the traditionally frame-by-frame implicit integrator problem into an optimization problem [13, 14]. These methods reduce the time spent on dataset preparation significantly, but the lack of ground truth means precise control over the training process is hard to achieve, and results are difficult to evaluate directly and quantitatively.

Graph neural networks (GNNs) have been successful in 3D data processing in recent years, with many approaches exploring the use of GNNs to solve garment mesh deformation tasks. The pioneering work in [43] introduced an innovative graph-learning method for auto skin binding of game characters, capable of predicting the skinning weights of complex skeletons using a unified model. This groundbreaking approach inspired a series of subsequent investigations [44–47], where different GNNs were used to learn skinning weights and character blend shapes. Researchers keen on replicating garment folds and wrinkles have adopted diverse architectures, such as the PointNet-based framework [48], graph attention networks (GATs) with output decomposition [36], and UNet-like GNNs [34]. These methodologies primarily mimic garment behavior on parametric human bodies based on SMPL. Extending beyond SMPL bodies, the study in [49] successfully handles non-SMPL objects and rigid bodies. However, the constraint remains that a trained model can only be applied to a fixed garment mesh topology. While these techniques are proficient in generating credible results for tight-fitting garments and static poses, they struggle with loose-fitting garments and dynamic effects. The introduction of temporal information as key data for the model to learn has been proposed as a solution to these challenges [11, 15, 50, 51]. The challenge persists in generating clear details while concurrently achieving dynamics, using a model that is generalizable and possesses robust inference capabilities. Our proposal extends beyond the conventional focus on classical GNNs, exploring the potential of graph transformers in garment deformation tasks.

3 Methodology

3.1 Neural simulation for garments

3D garment deformation can be regarded as a shift in the state of the garment mesh, wherein the intricate dependencies between each node and its surroundings naturally align with the strengths of GNNs. Furthermore, the topological agnosticism of these neural networks is particularly advantageous as it facilitates the application of learned models to a diverse array of garment types, regardless of their structural form. Despite the promise, as mentioned in many studies [10, 15, 36], straightforward application of this concept to neural cloth simulation tends to produce unsatisfactory results. To harness the full potential of GNNs, we suggest a novel approach in constructing graphs and refining the graph processing workflow, specifically tailored to the task of garment mesh deformation.

Our process begins by constructing a rest pose garment-body mesh graph. For the target garment $\mathbf{T}$ with N vertices, we create a base graph $\bar{\mathcal{G}}$, with each vertex described by a feature vector $\bar{\boldsymbol{x}}$ giving its spatial position, normal vector, and distances to skeletal joints, while edges represent vertex interconnections. This graph initially passes through an original transformer, which yields a transformed graph endowed with features $\bar{\boldsymbol{x}}^{[1]} \in \mathbf{R}^{N\times d}$ (where d is the number of feature channels). Concurrently, we compute the affinity matrix $A \in \mathbf{R}^{N\times N}$ for the garment, capturing geodesic distances between the N vertices. Applying the Nyström method to A enables us to approximate its eigenvectors $\boldsymbol{u} \in \mathbf{R}^{Z\times Z}$, with $Z \ll N$, thus simplifying the eigendecomposition. These eigenvectors are then processed through fully connected layers to generate features $\boldsymbol{u}^{[1]} \in \mathbf{R}^{1\times d}$. Subsequently, we perform element-wise multiplication of these features with the features of the transformed mesh graph: $\boldsymbol{x}' = \bar{\boldsymbol{x}}^{[1]} \odot \boldsymbol{u}^{[1]}$. This operation yields an enhanced graph characterized by the integrated features $\boldsymbol{x}'$. This integration serves two primary purposes. Firstly, it employs the fully connected layers as a selective filter, which emphasizes essential geodesic features while downplaying less critical ones, thereby endowing the model with a contextualized understanding of the garment mesh that enhances deformation precision. Secondly, it offers a comprehensive view that merges local vertex detail with broader structural patterns. Such an operation enhances the ability of the model to internalize the inherent geometry, which is particularly beneficial for adapting to changes in mesh topology.

To further infuse the model with body-specific features, high-dimensional features are extracted from the SMPL body model, parameterized by the shape coefficients β. This process, conducted through fully connected layers, generates features $\boldsymbol{b}^{[1]} \in \mathbf{R}^{d\times S}$ that quantify the distances from each vertex on the body mesh to each skeletal joint, where S represents the total number of joints. These extracted body features are then combined with the garment graph features $\boldsymbol{x}'$ through a multiplication operation, forming a unified static graph that represents the garment-body relationship. This process is described by $\boldsymbol{x}'' = \boldsymbol{x}' \otimes \boldsymbol{b}^{[1]}$, where $\boldsymbol{x}''$ represents the resulting feature-rich graph representation.

In processing dynamic information, we analyze pose histories Θ^t using a GRU to extract motion features $\Theta^{t[1]} \in \mathbf{R}^{1\times S}$ that are then integrated with the static garment graph: $\boldsymbol{x}''' = \boldsymbol{x}'' \odot \Theta^{t[1]}$. This fusion facilitates the creation of the skinning weights $\mathcal{W}^t$ and the blend shape B^t, where the blend shape is relatively complicated, and we employ additional transformers to generate it. The dynamic-aware garment mesh $M^{t,c}$ can be defined as:

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

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

where $W(\cdot)$ is the skinning function which deforms the unposed $T^t(\cdot)$ using the skinning weights $\mathcal{W}^t(\cdot)$ relative to joint locations $\mathbf{J}$. T^t is the result of the template garment mesh $\mathbf{T}$ combined with the time-specific blend shape B^t. Upon deriving $M^{t,c}$, we construct its corresponding dynamic-aware graph, embedding vertex features $\boldsymbol{x}$ and edge features $\boldsymbol{e}$. Vertex features incorporate the velocities between the previous and current frames, vertex normal of the current frame, and distances to all joints. Edge features include the lengths and normal of the edges.

Having constructed the dynamic-aware graph, our goal is to generate garment deformations with dynamic details. To achieve this, we leverage the capabilities of graph transformers, which are adept at handling the intricate dependencies and features within the graph data structure. Building on this foundation, we further enhance the model by our SAT design. The advanced model incorporates additional structural information into the transformer framework, allowing for a deeper understanding of the topology of the garment. Consequently, our model can precisely capture the way garments fold, stretch, and fit around the moving body, reflecting the subtle dynamics and wrinkles that conventional graph transformers may overlook. The SAT is engineered to output the acceleration of nodes in their current state. Integrating this calculated acceleration with the coarse garment mesh $M^{t,c}$ results in a garment deformation $M^{t,d}$ that captures dynamic details effectively. In essence, our

method is characterized by an information transfer process within a graph structure, with the processing conducted by the proposed graph-based transformer. Unlike systems constrained to fixed-size input vectors, this graph-based approach aligns with the demonstrated flexibility of other GNN-based simulations [15, 34, 41, 44], which do not necessitate predefined topology.

3.2 Graph transformer preliminaries

In this section, we provide a brief overview of the graph transformer structure [52, 53]. This structure can be thought of as a graph-based version of the original architecture [54]. The transformer uses a self-attention mechanism to recognize and encode the unique information found at different positions of graph nodes, creating output features for each location, which can be mathematically represented as:

$$\begin{aligned} \boldsymbol{q}_i &= \boldsymbol{W}_q \boldsymbol{x}_i^{[\ell]} + \boldsymbol{b}_q, \\ \boldsymbol{k}_j &= \boldsymbol{W}_k \boldsymbol{x}_j^{[\ell]} + \boldsymbol{b}_k, \\ \boldsymbol{h}_{ij} &= \frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{d_a}}, \\ \boldsymbol{\alpha}_{ij} &= \frac{\exp(\boldsymbol{h}_{ij})}{\sum_{u \in \mathcal{N}_i} \exp(\boldsymbol{h}_{iu})}, \end{aligned} \tag{3}$$

where $\boldsymbol{q}$ and $\boldsymbol{k}$ represents the queries and keys formed by learned linear transformations of the vertex features $\boldsymbol{x} \in \mathbf{R}^{N \times d}$ within a transformer layer. $\boldsymbol{x}^{[\ell]}$ stands for the vertex features in layer ℓ. $\boldsymbol{W}$ and $\boldsymbol{b}$ are used to represent corresponding weights and bias in a linear transformation, respectively. d_a is the number of feature channels, $\mathcal{N}_i$ is the first-order-neighbor vertex set of vertex i, and $\boldsymbol{\alpha}$ denotes the softmax attention scores, which are derived from the scaled dot product of queries and keys, passed through a softmax function for normalization. Although d_a and d can be set independently, in line with conventional practice, we set both to the same value in our method. The features of the following layer can then be generated by multiplying the values by these attention scores as follows:

$$\begin{aligned} \boldsymbol{v}_j &= \boldsymbol{W}_v \boldsymbol{x}_j^{[\ell]} + \boldsymbol{b}_v, \\ \hat{\boldsymbol{x}}_i^{[\ell+1]} &= \sum_{j \in \mathcal{N}_i} \boldsymbol{\alpha}_{ij} \boldsymbol{v}_j, \\ \boldsymbol{x}_i^{[\ell+1]} &= \boldsymbol{W}_o \hat{\boldsymbol{x}}_i^{[\ell+1]} + \boldsymbol{x}_i^{[\ell]}, \end{aligned} \tag{4}$$

where $\boldsymbol{v}$ represents the values within a transformer block, and $\boldsymbol{x}^{[\ell+1]}$ denotes the vertex features of the following layer. In the multi-headed case, the mid-level vertex features $\hat{\boldsymbol{x}}^{[\ell+1]}$, corresponding to different heads, are either averaged or concatenated prior to the linear transformation.

3.3 Structure-augmented transformer (SAT)

The connectivity between vertices, stored in edges, is just as important as vertex-level features in simulating clothing forces. Drawing inspiration from a previous study [53] that introduced edge processing into graph transformers, we incorporate garment structure information into the attention calculation and feature aggregation. This extends the original transformer architecture, taking advantage of the transformer's ideal permutation equivariance properties for processing graph data: the layer is invariant to permutations of vertices in a graph, provided the edges remain consistent. This property makes transformers suitable for concurrently processing vertex and edge features, given that graphs themselves are invariant to node ordering, provided the connectivity remains consistent. For our case, the initial edge features $\boldsymbol{e}^0$ are defined using displacements and directions of two adjacent frames, and then are forwarded into transformer layers to generate the ℓ-th edge features $\boldsymbol{e}^{[\ell]} \in \mathbf{R}^{N \times N \times d_e}$. Here, N denotes the number of vertices, and d_e is the number of channels of edge features. The edge features are propagated across each layer, all the way up to the network output.

As given by Eqs. (3) and (4), the attention scores can be considered a normalized adjacency matrix provided by a weighted complete graph. These scores determine how the values $\boldsymbol{v}$ are aggregated to form subsequent layer features. Unlike the input graph which is manually defined, the graph features in intermediate layers are adaptively formed by the self-attention mechanism. However, the original transformer does not offer a direct method to incorporate the garment graph structure into feature propagation. Moreover, these hidden graph features are immediately collapsed when aggregation is finished. To address this challenge, we allow the edge features to participate in the attention calculation as feature gating. The hidden edge features $\boldsymbol{e}$ are linearly transformed and scaled by the sigmoid function $\sigma(\cdot)$, enabling the edge information to gate the attention scores and thereby regulate the information flow between vertices. Our attention calculation is as follows:

$$\begin{aligned} \boldsymbol{p}_{ij} &= \boldsymbol{W}_p \boldsymbol{e}_{ij}^{[\ell]} + \boldsymbol{b}_p, \\ \boldsymbol{q}_{ij} &= \boldsymbol{W}_q \boldsymbol{e}_{ij}^{[\ell]} + \boldsymbol{b}_q, \\ \boldsymbol{h}_{ij} &= \frac{\boldsymbol{q}_i^T \boldsymbol{k}_j}{\sqrt{d_a}} \sigma\left(\boldsymbol{p}_{ij}\right), \\ \boldsymbol{\alpha}_{ij} &= \frac{\exp(\boldsymbol{h}_{ij})}{\sum_{u \in \mathcal{N}_i} \exp(\boldsymbol{h}_{iu})}, \end{aligned} \tag{5}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the hidden edge features formed by learned linear transformations of the edge features $\boldsymbol{e}$. For aggregation, we enhance the representation of the graph structure by

combining the values $\boldsymbol{v}$ and edge features $\boldsymbol{q}$ as follows:

$$\begin{aligned}\hat{\boldsymbol{x}}_i^{[\ell+1]} &= \sum_{j\in\mathcal{N}_i} \boldsymbol{\alpha}_{ij}\left(\boldsymbol{v}_j + \boldsymbol{q}_{ij}\right),\\ \boldsymbol{x}_i^{[\ell+1]} &= \boldsymbol{W}_o\hat{\boldsymbol{x}}_i^{[\ell+1]} + \boldsymbol{x}_i^{[\ell]},\\ \boldsymbol{e}_{ij}^{[\ell+1]} &= \boldsymbol{W}_e\boldsymbol{h}_{ij} + \boldsymbol{e}_{ij}^{[\ell]},\end{aligned} \tag{6}$$

where $\boldsymbol{v}$ is computed in the same way as in the original definition. To change the number of channels, we also add a linear transformation on the $\boldsymbol{x}_i^{\ell}$ in a residual connection. The integration of edge features presents a practical controlling capacity for the information spread among vertices, leading to better and faster convergence during training. To ensure the network capitalizes on the introduced connectivity, we also randomly add $-\infty$ to the softmax during training to induce a small probability for attention masking. Typically, the output is processed by a subsequent LayerNorm and feed-forward module [53]. In practice, we replace this module by straightforward ELU activation: we find that this adjustment neither hinders the results nor affects training, but it speeds up inferencing.

3.4 Unified deformation network optimization

Our optimization strategy comprises both supervised and unsupervised components. Differing from purely supervised or unsupervised methods, we integrate ground-truth-based supervised losses and physics-informed unsupervised losses to achieve unified optimization for the garment deformation network. Our loss terms are borrowed from multiple previous studies [13, 14, 41, 48, 55], with certain modifications to suit our unified training strategy.

We use vertex position [41], edge length, and norm angle [48] for the supervised loss terms:

$$\mathcal{L}_{\text{vert}} = k_v\frac{1}{N}\sum_i^N \|\boldsymbol{p}_i - \boldsymbol{p}_i^{\text{GT}}\|_2, \tag{7}$$

$$\mathcal{L}_{\text{edge}} = k_e\|\boldsymbol{E} - \boldsymbol{E}^{\text{GT}}\|_2^2, \tag{8}$$

$$\mathcal{L}_{\text{norm}} = k_n\frac{1}{N}\sum_i^N(\boldsymbol{1} - \boldsymbol{n}_i^T\boldsymbol{n}_i^{\text{GT}}), \tag{9}$$

where k_v, k_e, and k_n are hyperparameters for balancing these losses, respectively. $\boldsymbol{p}$ denotes the predicted vertex positions. $\boldsymbol{E}$ denotes edge lengths in the deformed garment (to distinguish them from edge features in the previous section, we use a capital $\boldsymbol{E}$ here). $\boldsymbol{n}$ denotes vertex normals. The superscript GT indicates ground truth data.

Building on the supervised losses, we further incorporate physics-informed unsupervised losses as auxiliary components. This strategy is useful for training a dataset with a wide variety of garments and complex motions, and can enhance the network's ability to generalize to unseen garments or poses. Specifically, we include collisions [13], gravity [14], stretch forces, and shearing forces [55] in our losses as follows:

$$\mathcal{L}_{\text{collision}} = k_c\sum_i^N \left|\min\left(s(\boldsymbol{p}_i) - \epsilon, 0\right)\right|, \tag{10}$$

$$\mathcal{L}_{\text{gravity}} = -k_g\sum_i^N \boldsymbol{m}_i\boldsymbol{p}_i^T\boldsymbol{g}, \tag{11}$$

where k_c, k_g, k_s, and k_h are hyperparameters for balancing unsupervised losses. The function $s(\cdot)$ calculates the signed distance from a garment vertex to its corresponding body mesh. ϵ is a collision threshold value used to increase robustness. $\boldsymbol{m}_i$ represents the vertex mass, which is computed from vertex area and fabric surface density. $\boldsymbol{g} = [0, 0, -9.8]$ denotes gravitational acceleration. To better simulate the cloth model, stretch and shearing forces are used as follows:

$$\mathcal{L}_{\text{stretch}} = k_s\sum_i^{N_F} \boldsymbol{a}_i\left(\|\boldsymbol{W}_i^u\|_2 - 1\right)^2 + \boldsymbol{a}_i(\|\boldsymbol{W}_i^v\|_2 - 1)^2, \tag{12}$$

$$\mathcal{L}_{\text{shear}} = k_h\sum_i^{N_F} \boldsymbol{a}_i\left(\boldsymbol{W}_i^u\boldsymbol{W}_i^v\right)^2, \tag{13}$$

$$(\boldsymbol{W}^u\boldsymbol{W}^v) = \begin{pmatrix}\Delta\boldsymbol{p}_1 & \Delta\boldsymbol{p}_2\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{u}_1 & \Delta\boldsymbol{u}_2\\ \Delta\boldsymbol{v}_1 & \Delta\boldsymbol{v}_2\end{pmatrix}^{-1}, \tag{14}$$

where $\boldsymbol{a}_i$ is the i-th triangle's area in uv coordinates and N_F is the number of faces. For $\mathcal{L}_{\text{stretch}}$ and $\mathcal{L}_{\text{shear}}$, let us first consider a face of the garment mesh, indexed by vertex i, j, k. We then define $\Delta\boldsymbol{p}_1 = \boldsymbol{p}_j - \boldsymbol{p}_i$ and $\Delta\boldsymbol{p}_2 = \boldsymbol{p}_k - \boldsymbol{p}_i$. Similarly, we set $\Delta\boldsymbol{u}_1 = u_j - u_i$, $\Delta\boldsymbol{u}_2 = u_k - u_i$, and the same applies to $\Delta\boldsymbol{v}_1, \Delta\boldsymbol{v}_2$. The vertex position $\boldsymbol{p}_i$ changes in world space, and the fixed plane coordinate is represented as $(\boldsymbol{u}_i, \boldsymbol{v}_i)$. As in physical simulations, these losses can effectively regulate stretch anisotropically in deformation generation.

During training, we initially fine-tune the network with supervised losses to ensure it learns the fundamental deformations according to the ground truth. As the predictions become increasingly accurate and the rate of improvement plateaus, we transition to unsupervised learning to polish the performance of the model. Details of the training setup are outlined in Sec. 4.1. The proposed unified optimization strategy helps us to mitigate the issues associated with purely supervised learning, such as the lack of physical consistency, as well as the substantial deviation toward ground truth data found in purely unsupervised learning. It is important to note that minor deviations from the ground truth do not undermine the efficacy of the method. In real-world applications, ground truth data is often unavailable, and the perceived quality of deformations is paramount. Our method allows for a better

balance between accuracy and realism, bridging the gap between theoretical models and practical utility.

4 Experiments

4.1 Experimental setup

To train our network, we first amassed a garment dataset that includes various garment types. These were collected from the CLOTH3D dataset and draped over SMPL bodies of differing shapes. To animate these garments, we extracted motion sequences from the CMU Mocap dataset and CLOTH3D, with the frame rate set to 30. For garment simulation, we used the physics model within Blender, using silk-like fabric settings. The training set comprises approximately 50,000 poses, each associated with 50 garments and three different body shapes. The body shapes were deliberately sampled within the SMPL parameter domains, primarily the first body shape parameter, corresponding to thin, regular, and fat respectively. Our validation dataset comprises 5 body-garment pairs with 2,500 poses. The test dataset includes around 5,000 poses, associated with 15 different garments draped over randomly-generated bodies.

Our network consists of two distinct modules: (i) the dynamic-aware graph construction module, and (ii) the dynamic detail deformation generation module.

In the graph construction module, rest-pose garment-body information was initially processed through several fully-connected (FC) layers and transformer layers. The FC layers for garment processing contain [128, 128, 128] hidden channels and utilize tanh activation. The transformer layers dedicated to rest-pose garment graph processing contain [64, 64, 128] hidden channels and four heads, also utilizing tanh activation. The input graph was composed of vertices, normals, and distances to body joints. The FC layers for body processing contained [512, 256, 128] hidden channels and used tanh activation. The input body feature was generated using distances between body vertices and body joints. We used a two-layer gate recurrent unit (GRU) with 128 hidden channels to process motion sequences, and fused dynamic encodings with the processed rest-pose information.

The fused features, further processed by a softmax function, were used as skinning weights. The fused features forwarded into additional transformer layers were used to generate blend shapes. These transformer layers contained [32, 32, 32, 3] hidden channels and four heads, with ELU activation utilized for all except the output. The outputs were used to construct the dynamic-aware graph, which consists of two types of features: vertex- and edge-level features. The vertex-level features included the velocity between the current and previous frames, the vertex normal, and the distances from a vertex to body joints. The edge-level features included the length of an edge and the direction of two vertices of an edge.

Moving to the dynamic detail deformation generation module, we used six SATs with [32, 64, 64, 64, 32, 3] hidden channels and four heads in series to process the dynamic-aware graph. We chose ELU activation for SATs based on empirical observation that an activation with a negative saturation region has a more stable effect on acceleration prediction. For the intermediate SAT layers where the number of channels changes, we introduced an additional linear transformation on the residual connection to adjust the number of hidden features. The predicted accelerations were generated by averaging the final four-head results. Predicting acceleration, an idea inspired by [15], is advantageous in this context due to its relative nature, effectively circumventing the need for direct prediction of the absolute position of the deformation result.

For parameter initialization, we used the geometric initialization method from [56] for the linear layers. Meanwhile, the parameters within the GRU and Transformers were initialized using the Kaiming initialization method [57].

Given that we integrate multiple loss terms for our network's efficacy, we used a progressive training mode to ensure stable convergence. Initially, we updated the parameters solely using the supervised losses, including $\mathcal{L}_{\text{vert}}$, $\mathcal{L}_{\text{edge}}$, and $\mathcal{L}_{\text{norm}}$. Once the vertex error reduction rate fell below 3% within 50 epochs, we activated $\mathcal{L}_{\text{collision}}$ and $\mathcal{L}_{\text{gravity}}$, while scaling the supervised losses by a factor of 0.05. Upon stabilization of collision error, we froze the rest-pose processing submodule and activated the remaining losses, $\mathcal{L}_{\text{stretch}}$ and $\mathcal{L}_{\text{shear}}$. The hyperparameters for supervised losses k_v, k_e, and k_n were set to 100, 15, and 50, respectively. The hyperparameters for the unsupervised losses k_c, k_g, k_s, and k_h were set to 1.5, 0.1, 5, and 2, respectively. We employed the Adamax optimizer [58], with an initial learning rate of .003, and applied cosine annealing to decay the learning rate. Network training was conducted on two servers, one equipped with two nVIDIA RTX A6000 GPUs and the other with two H800 GPUs.

For testing, we utilized a computer with an Intel i9 13900K CPU and an nVIDIA GeForce RTX 4090 GPU. Our method achieved inference speeds of about 29ms per frame for garments with an average vertex count of 10K. For garments with the highest vertex count in our dataset, about 14K vertices, the inference time was around 33ms per frame. Although an increase in vertex count should theoretically have only a small impact on runtime, practical performance can be

Fig. 2 Qualitative comparison of garment deformations with and without edge processing. Models that incorporate structural information can precisely simulate garment deformation, resulting in more vivid and realistic outcomes.

Fig. 3 Qualitative comparison of garment deformations with different edge processing strategies. Our method yields more refined and satisfactory details than alternative approaches.

compromised by several factors. Specific characteristics of CUDA, cuDNN, and PyTorch versions used, such as inefficient memory management and suboptimal kernel optimizations, may extend computation times. Additionally, performance may be impacted by uneven resource allocation during GPU parallel processing, especially when computational tasks are not uniformly distributed across cores. Despite this, our approach consistently maintains high inference speeds, and supports batch processing, effortlessly meeting real-time requirements.

4.2 Evaluation

4.2.1 Structure-augmented transformer

To demonstrate the effectiveness of the proposed SAT module, we investigated the impact of integrating edge information within the transformer framework. Fig. 2 presents a comparative analysis of garment deformation outcomes for our method with edge processing, and the original graph transformer approach without edge processing. In the top row, garments lacking edge processing show a loss of critical structural constraints, leading to deformations that diverge from the natural folds of clothing. Notably, artifacts predominantly arise near the body joints, which are crucial interaction points between the garment and body. Without edge processing, the model may inaccurately simulate movements and bending at these joints, resulting in artifacts when the fabric should exhibit stretch or compression. In contrast, the SAT leverages the mesh structural information provided by the edges to achieve garment deformations showing more natural and realistic results.

Additionally, we conducted a validation of the impact of various edge processing techniques on the visual appearance of garments. Fig. 3 compares three strategies: the approach without edge gating as in [53], the use of a structure-aware transformer architecture [59], and our proposed method. These strategies were tested across a range of representative garments subjected to various bodies and movements. When edge gating was absent, the model could fail to differentiate between structurally critical and non-critical edge interactions, often glossing over the intricate details that contribute to realistic garment appearance, leading to less detailed output. The structure-aware transformer, while showing some improvement, also fell short in certain areas. Despite its general effectiveness in graph representation learning, it appears to lack the specificity required for accurately simulating the complex behaviors of garment deformation. This was evident in its handling of the trouser part of the jumpsuits and the right sleeve portions of the second and third blue dresses, where fine details were not fully realized. In contrast, our proposed method benefits from the our edge processing that allows for a more detailed representation of the context-dependent behaviors of garments, resulting in a visibly superior level of detail.

4.2.2 Network optimization

To assess our unified network optimization strategy for garment deformation, we conducted experiments by excluding the unsupervised loss components; results are presented in Fig. 4. Specifically, we removed stretch and shearing losses while maintaining collision and gravity losses. (Omitting collision and gravity losses produced many collisions and yielded garment behaviors that grossly violated physical laws, so these losses were retained to ensure basic physical realism). The displayed outcomes indicate that our network was able to preserve the general form of garments despite the absence of stretch and shearing losses, showing its inherent robustness. Nonetheless, garments appeared unnaturally rigid and lacked high-frequency details without these losses, compromising the quality of deformation, especially for loose garments like dresses. This highlights the important role of stretch and shearing losses in capturing the detailed behavior of fabric and achieving realistic garment simulations. Conversely, when we experimented with removing the supervised loss and relied solely on the unsupervised loss, this led to non-convergence of the network, preventing the emergence of viable deformation results. Supervised loss is critical for aligning network predictions with precise ground truth data, as necessary for complex deformation tasks. Without it, the network lacks the necessary guidance to capture the intricacies of garment behavior. This experimentation highlights the complementary interplay between supervised and unsupervised losses within our optimization framework. The combination of these losses is indispensable; the supervised loss directs the learning progress towards accurate deformation, while the unsupervised loss upholds the physical realism of the deformations. Therefore, their combination is crucial for the network to yield simulations that are not only stable and convergent but also visually convincing.

4.2.3 Quantitative evaluation

Correspondingly, we also quantitatively evaluated the predicted results from various transformer implementations and loss optimization strategies as outlined in Table 1. We considered test garments of two types: jumpsuits and dresses. For each category, we compared the predictions of different method variants to the ground truth, using five distinct metrics to assess performance. Our analysis reveals that edge

Fig. 4 Qualitative comparison of deformations with and without unsupervised losses ($\mathcal{L}_{\text{stretch}}$ and $\mathcal{L}_{\text{shear}}$). Test garments include both tight-fitting jumpsuits and loose-fitting skirts. Removing unsupervised losses leads to a loss of detail, resulting in garments that appear stiffer.

Table 1 Comparison of different transformer implementations and loss optimization strategies. Predictions are evaluated against ground truth using five metrics: average vertex distance E_{dist} (cm), average deviation of vertex normal E_{vnorm} (°), and face normal E_{fnorm} (°), relative edge length error E_{len} (%), and discrete Gaussian curvature error E_{curv}.

Metric / Method	Jumpsuit					Dress				
	E_{dist}	E_{vnorm}	E_{fnorm}	E_{len}	E_{curv}	E_{dist}	E_{vnorm}	E_{fnorm}	E_{len}	E_{curv}
w/o edge processing	3.39	20.43	22.25	11.84	0.043	3.82	22.83	25.69	12.07	0.057
w/o edge gating	2.81	18.35	21.82	9.56	0.039	3.21	20.08	22.15	10.93	0.043
structure-aware transformer	2.92	17.14	19.49	9.87	0.034	3.13	20.77	22.51	11.43	0.038
w/o unsupervised loss	2.72	15.25	17.91	10.31	0.031	3.37	21.09	23.68	9.15	0.045
ours	**2.13**	**13.58**	**16.41**	**7.35**	**0.026**	**2.45**	**17.27**	**19.53**	**8.94**	**0.031**

processing significantly influences the numerical outcomes, being particularly impactful. Additionally, the presence of unsupervised loss plays a crucial role, especially in the accurate deformation of dresses. These findings underscore the importance of both edge processing techniques and unsupervised loss components in the precise simulation of garment deformations. Our method consistently outperforms alternative approaches across all evaluated metrics, achieving the lowest error. This performance demonstrates the efficacy of our method and establishes it as a superior solution for garment deformation simulations.

4.2.4 Details and dynamics in continuous motion

The simulation of loose-fitting garments, such as dresses, poses distinct challenges due to their non-conformal behavior relative to the body, leading to complex deformations. The dynamic state of movement for such garments introduces individualized effects that can vary greatly. In Fig. 5, we showcase the effectiveness of our method using three test dresses, each with a distinct body shape and motion pattern. Our approach successfully captures the realistic behavior of the hemlines, for various hemline styles undergoing movement. The dynamic effect of the swing of the dress is tied to both the characteristics of the garment and the amplitude of the movement. For example, when there is a rapid and large rotation of the skeletal joints, the dress responds with a pronounced dynamic sway. Conversely, when the rotation is slower, the dynamic effect is accordingly muted, resulting in a more gentle sway. This variance in dress response is reflective of the physically realistic behavior we observe.

Additionally, in Fig. 6, our approach demonstrates its ability to predict dynamic deformations of dresses over long motion sequences. Owing to our strategically designed method for the effective transfer of dynamic graph information and a holistic network optimization process, our system maintains its ability to generate physically consistent and realistic

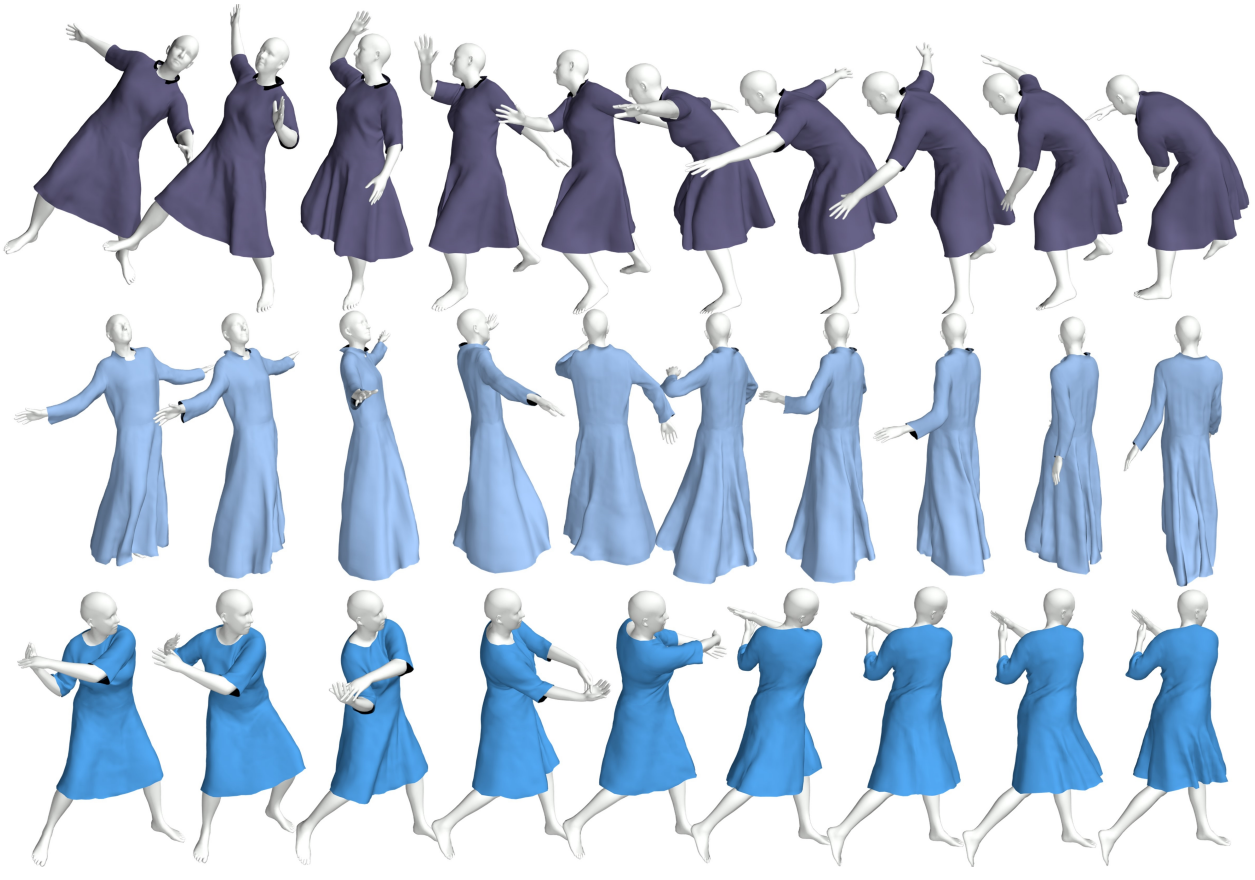


Fig. 5 Our method adeptly simulates dynamic effects and intricate details for a variety of unseen loose-fitting garments, body shapes, and motion scenarios.

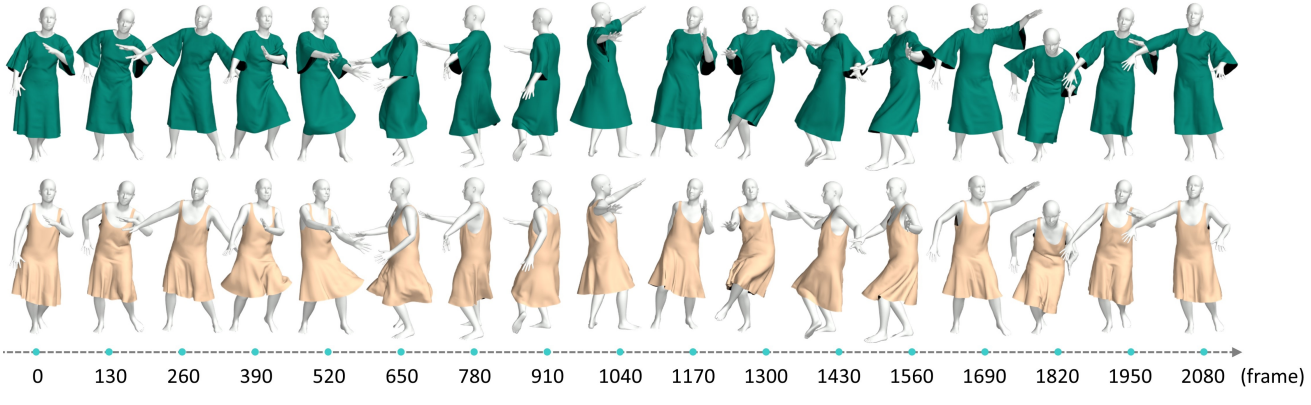


Fig. 6 Our method demonstrates a robust ability to approximate garment dynamics across long motion sequences, consistently reproducing deformations with rich folds and detailed wrinkles.

garment dynamics throughout lengthy sequences.

4.3 Comparison

4.3.1 Capacity comparison

To further evaluate the capabilities of our proposed methodology, we conducted a comparative analysis with the most recent learning-based clothing deformation techniques: NCS [13], HOOD [15], and DA [36]. Table 2 details the key features of these methods. NCS exhibits strong capabilities in physical simulation but is limited to handling specific types of clothing; even slight increases in the number of vertices can prevent it from generating deformations. This indicates that NCS lacks scalability for garments with varying vertex counts and robustness in dealing with different types of clothing. In contrast, HOOD can process a variety of clothing types and topologies, and it generates results that

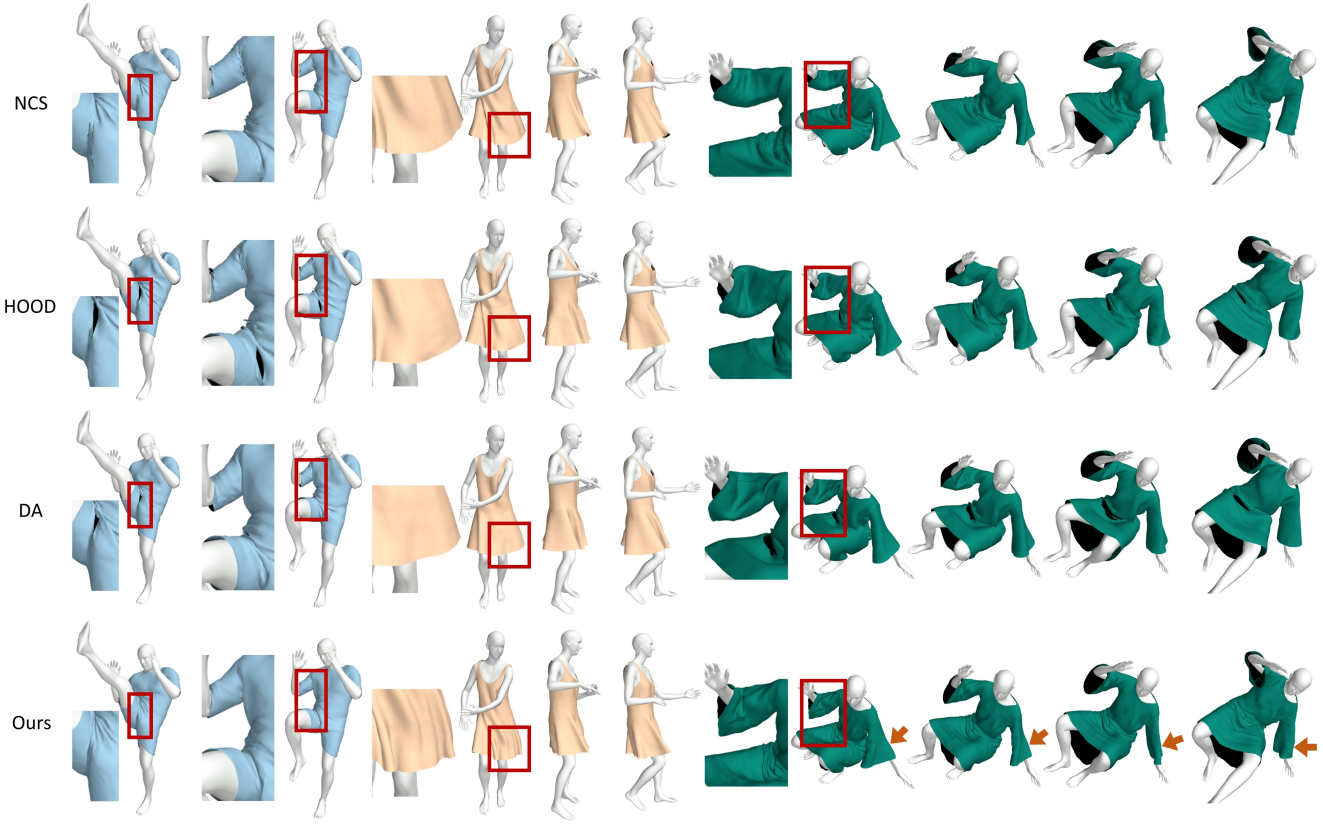


Fig. 7 Qualitative comparison to state-of-the-art approaches NCS, HOOD, and DA. Our predictions display reasonable dynamics and rich details.

Table 2 Comparison of our method to state-of-the-art approaches: NCS, HOOD, and DA, highlighting differences in deformation handling, learning schemes, garment generalization capabilities, and batch processing support.

	Deformation type	Learning scheme	Genera-lization	Batch support
NCS	dynamic	unsupervised	✗	✓
HOOD	dynamic	unsupervised	✓	✗
DA	static	supervised	✓	✓
Ours	dynamic	unified	✓	✓

closely mimic physical simulation effects, demonstrating good scalability and robustness. DA also shows strong scalability and robustness; however, its trained model only considers static deformations and lacks unsupervised constraints for dynamic effects such as gravity and inertia, which impacts the realism of the deformation outcomes. Our method effectively combines the advantages of both supervised and unsupervised learning to simulate dynamic garment deformations, which is particularly beneficial for animating digital characters dressed in loose-fitting clothing. Additionally, our method demonstrates a robust generalization capability, and is adept at predicting deformations for arbitrary garments, body shapes, and motions, enabling accurate garment deformation prediction without the necessity for retraining. This flexibility is essential for managing diverse character conditions and can significantly reduce processing time.

The NCS model focuses on one specific garment, allowing it to operate using a simple architecture of linear layers and nonlinear activations, achieving an average speed of 853 fps. In contrast, our method and DA, when processing batches of data in parallel, reach inference speeds of 280 fps and 467 fps, respectively. Although these speeds ensure real-time performance, they are significantly slower than NCS. On the other hand, HOOD requires the inclusion of the previous frame's computation results when calculating the next frame, so is subject to a single-batch limitation. Consequently, HOOD's actual inference speed is approximately 14 fps, much slower than the other methods. Overall, our approach offers a more comprehensive solution and greater practicality for real-world applications.

4.3.2 *Qualitative comparison*

We conducted a qualitative comparison of our method to state-of-the-art approaches, as depicted in Fig. 7, which demonstrates garment deformation results for jumpsuits and dresses during various unseen movements such as

Table 3 Quantitative comparison to state-of-the-art approaches. Five metrics are used for comparison: average vertex distance E_{dist} (cm), average deviation in vertex normal E_{vnorm} (°) and face normal E_{fnorm} (°), relative edge length error E_{len} (%), and discrete Gaussian curvature error E_{curv}.

	E_{dist}	E_{vnorm}	E_{fnorm}	E_{len}	E_{curv}
NCS	4.93	29.43	33.08	14.93	0.072
HOOD	6.48	33.85	36.52	16.27	0.089
DA	7.09	34.26	37.14	16.19	0.104
Ours	**2.36**	**17.38**	**18.92**	**8.69**	**0.029**

kicking, rotating, and hand-supported standing-up. Given the tighter fit of the jumpsuit than the dress, we focused on the fidelity of wrinkle detail reproduction. The NCS model, tailored to specific garments, achieves reasonably accurate deformations. Conversely, the HOOD and DA models, which boast generalizability across different garments, tend to exhibit less natural deformation effects due to their limited ability to capture structural garment information. Our method stands out by producing more refined details with an absence of noticeable artifacts. In the case of the orange dress undergoing a rotating movement, it is crucial for the hemline to exhibit a dynamic, swinging effect synchronized with the movement. The DA approach falls short in capturing this dynamic, functioning primarily as a static model. The HOOD and NCS methods manifest a degree of dynamism but do not offer the rich, lifelike folds seen in our output. For the dark green dress featuring loose sleeves, our method captures the swinging motion during the hand-supported stand-up, highlighted by the arrows in the figure. The resulting dynamic is not only convincing but also includes the formation of natural-looking wrinkles, showcasing the effectiveness of the SAT and unified learning scheme. Although our method necessitates a certain amount of preparation time to establish the ground truth, unlike purely unsupervised approaches, this investment enables our model to learn and replicate clothing behavior with high accuracy. This trade-off is justified by the enhanced deformation realism our method achieves.

4.3.3 Quantitative comparison

We provide a quantitative comparison in Table 3 to further compare the consistency between each method and the ground truth. Each of the five metrics employed demonstrates enhanced accuracy in our predictive models. Nonetheless, we must acknowledge that this is partially because NCS and HOOD use unsupervised learning, which can lead to numerical bias. Additionally, while DA yields results that are numerically close to those of HOOD, they do not necessarily match in terms of visual dynamic equivalence. As highlighted in [42], users are typically unaware of the ground truth, which makes numerical biases difficult to discern. Consequently, we argue that qualitative assessments should be prioritized.

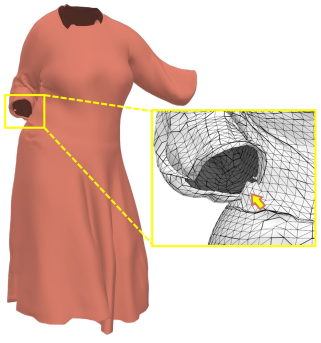


Fig. 8 Example of self-collision. Clenching of the shoulder joint brings the arm into close contact with the torso, leading to mesh interpenetration between the arm and upper body. This self-collision challenge is not yet comprehensively managed by our existing framework.

5 Conclusions

We have introduced a novel learning-based approach, GarTrans, capable of realistically deforming various garments on different body shapes performing diverse motions. Our methodology is composed of two pivotal modules that work in tandem to produce the deformations. The dynamic-aware graph construction module incorporates multi-source deformation factors into the graph structure, providing a comprehensive representation of the dynamic state of the garment mesh. Subsequently, the dynamic detail deformation generation module utilizes the SAT that allows nodes to effectively assimilate contextual information, thereby facilitating the generation of intricate folds and wrinkles with enhanced detail. Moreover, by harmonizing the benefits of both supervised and unsupervised learning, we provide a unified deformation optimization process which enhances the generation of high-fidelity garment deformations. Notably, our approach enables real-time performance that is roughly 20 times faster than physically-based simulators, paving the way for interactive applications. Currently, our collision constraints are specifically tailored to address interactions between the body and garment, which leaves room for improvement in scenarios where the clothing experiences self-collision (as in Fig. 8). Future work will focus on integrating self-collision state prediction [60] within our framework, aiming to enhance the quality of garment simulations.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 62402021, 62403017) and

Beijing Natural Science Foundation (Grant Nos. 4244088, 4232017).

Declaration of competing interest

The authors have no competing interests to declare relevant to the content of this article.

References

[1] Nealen A, Müller M, Keiser R, Boxerman E, Carlson M. Physically Based Deformable Models in Computer Graphics. *Comput. Graph. Forum*, 2006, 25(4): 809–836, doi:https://doi.org/10.1111/j.1467-8659.2006.01000.x.

[2] Narain R, Samii A, O'Brien JF. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph.*, 2012, 31(6), doi:10.1145/2366145.2366171.

[3] Li J, Daviet G, Narain R, Bertails-Descoubes F, Overby M, Brown GE, Boissieux L. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph.*, 2018, 37(4), doi:10.1145/3197517.3201308.

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

[5] Lewis JP, Matt C, Nickson F. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proc. Annu. Conf. Comput. Graph. Interact. Tech.*, 2000, 165–172, doi:10.1145/344779.344862.

[6] Lähner Z, Cremers D, Tung T. DeepWrinkles: Accurate and Realistic Clothing Modeling. In *Proc. Eur. Conf. Comput. Vis.*, 2018, 1–18, doi:10.1007/978-3-030-01225-0_41.

[7] Zhang M, Ceylan D, Mitra NJ. Motion Guided Deep Dynamic 3D Garments. *ACM Trans. Graph.*, 2022, 41(6), doi:10.1145/3550454.3555485.

[8] Zhang M, Wang TY, Ceylan D, Mitra NJ. Dynamic Neural Garments. *ACM Trans. Graph.*, 2021, 40(6), doi:10.1145/3478513.3480497.

[9] Santesteban I, Otaduy MA, Casas D. Learning-Based Animation of Clothing for Virtual Try-On. *Comput. Graph. Forum*, 2019, 38(2): 355–366, doi:https://doi.org/10.1111/cgf.13643.

[10] Patel C, Liao Z, Pons-Moll G. TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, 7363–7373, doi:10.1109/CVPR42600.2020.00739.

[11] Pan X, Mai J, Jiang X, Tang D, Li J, Shao T, Zhou K, Jin X, Manocha D. Predicting Loose-Fitting Garment Deformations Using Bone-Driven Motion Networks. In *ACM SIGGRAPH*, 2022, 1–10, doi:10.1145/3528233.3530709.

[12] Santesteban I, Thuerey N, Otaduy MA, Casas D. Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, 11758–11768, doi:10.1109/CVPR46437.2021.01159.

[13] Bertiche H, Madadi M, Escalera S. Neural Cloth Simulation. *ACM Trans. Graph.*, 2022, 41(6), doi:10.1145/3550454.3555491.

[14] Santesteban I, Otaduy MA, Casas D. SNUG: Self-Supervised Neural Dynamic Garments. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, 8130–8140, doi:10.1109/CVPR52688.2022.00797.

[15] Grigorev A, Thomaszewski B, Black MJ, Hilliges O. HOOD: Hierarchical Graphs for Generalized Modelling of Clothing Dynamics. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, 16965–16974, doi:10.1109/CVPR52729.2023.01627.

[16] Baraff D, Witkin A. Large Steps in Cloth Simulation. In *ACM SIGGRAPH*, 1998, 43–54, doi:10.1145/280814.280821.

[17] Li Y, Du T, Wu K, Xu J, Matusik W. DiffCloth: Differentiable Cloth Simulation with Dry Frictional Contact. *ACM Trans. Graph.*, 2022, 42(1), doi:10.1145/3527660.

[18] Liang J, Lin MC, Koltun V. Differentiable Cloth Simulation for Inverse Problems. In *Proc. Adv. Neural Inf. Process. Syst.*, 2019, 1–10.

[19] Bridson R, Fedkiw R, Anderson J. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph.*, 2002, 21(3): 594–603, doi:10.1145/566654.566623.

[20] Tang M, Tong R, Wang Z, Manocha D. Fast and Exact Continuous Collision Detection with Bernstein Sign Classification. *ACM Trans. Graph.*, 2014, 33(6), doi:10.1145/2661229.2661237.

[21] Wang B, Ferguson Z, Schneider T, Jiang X, Attene M, Panozzo D. A Large-Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection. *ACM Trans. Graph.*, 2021, 40(5), doi:10.1145/3460775.

[22] Tang M, Wang T, Liu Z, Tong R, Manocha D. I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation. *ACM Trans. Graph.*, 2018, 37(6), doi:10.1145/3272127.3275005.

[23] Li M, Kaufman DM, Jiang C. Codimensional Incremental Potential Contact. *ACM Trans. Graph.*, 2021, 40(4), doi:10.1145/3450626.3459767.

[24] Jiang C, Gast T, Teran J. Anisotropic Elastoplasticity for Cloth, Knit and Hair Frictional Contact. *ACM Trans. Graph.*, 2017, 36(4), doi:10.1145/3072959.3073623.

[25] Müller M, Heidelberger B, Hennix M, Ratcliff J. Position Based Dynamics. *J. Vis. Commun. Image Representation*, 2007, 18(2): 109–118, doi:10.1016/j.jvcir.2007.01.005.

[26] Müller M. Hierarchical Position Based Dynamics. In *Workshop Virtual Real. Interact. Phys. Simul.*, 2008, 1–10, doi:10.2312/PE/vriphys/vriphys08/001-010.

[27] Müller M, Chentanez N, Kim TY, Macklin M. Strain Based Dynamics. In *ACM SIGGRAPH Symp. Comput. Animat.*, 2014, 149–157, doi:10.2312/sca.20141133.

[28] Li C, Tang M, Tong R, Cai M, Zhao J, Manocha D. P-Cloth: Interactive Complex Cloth Simulation on Multi-GPU Systems Using Dynamic Matrix Assembly and Pipelined

Implicit Integrators. *ACM Trans. Graph.*, 2020, 39(6), doi: 10.1145/3414685.3417763.

[29] Wu L, Wu B, Yang Y, Wang H. A Safe and Fast Repulsion Method for GPU-Based Cloth Self Collisions. *ACM Trans. Graph.*, 2020, 40(1), doi:10.1145/3430025.

[30] Wu B, Wang Z, Wang H. A GPU-Based Multilevel Additive Schwarz Preconditioner for Cloth and Deformable Body Simulation. *ACM Trans. Graph.*, 2022, 41(4), doi: 10.1145/3528223.3530085.

[31] Chen Z, Chen HY, Kaufman DM, Skouras M, Vouga E. Fine Wrinkling on Coarsely Meshed Thin Shells. *ACM Trans. Graph.*, 2021, 40(5), doi:10.1145/3462758.

[32] Wang H. GPU -based Simulation of Cloth Wrinkles at Submillimeter Levels. *ACM Trans. Graph.*, 2021, 40(4), doi: 10.1145/3450626.3459787.

[33] Gillette R, Peters C, Vining N, Edwards E, Sheffer A. Real-Time Dynamic Wrinkling of Coarse Animated Cloth. In *ACM SIGGRAPH Symp. Comput. Animat.*, 2015, 17–26, doi:10.1145/2786784.2786789.

[34] Vidaurre R, Santesteban I, Garces E, Casas D. Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *Comput. Graph. Forum*, 2020, 39(8): 145–156, doi: https://doi.org/10.1111/cgf.14109.

[35] Tiwari G, Bhatnagar BL, Tung T, Pons-Moll G. SIZER: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing. In *Proc. Eur. Conf. Comput. Vis.*, volume 12348, 2020, 1–18, doi:10.1007/978-3-030-58580-8_1.

[36] Li T, Shi R, Kanai T. Detail-Aware Deep Clothing Animations Infused with Multi-Source Attributes. *Comput. Graph. Forum*, 2023, 42(1): 231–244, doi:https://doi.org/10.1111/cgf.14651.

[37] Wang TY, Shao T, Fu K, Mitra NJ. Learning an Intrinsic Garment Space for Interactive Authoring of Garment Animation. *ACM Trans. Graph.*, 2019, 38(6), doi: 10.1145/3355089.3356512.

[38] Chen L, Gao L, Yang J, Xu S, Ye J, Zhang X, Lai YK. Deep Deformation Detail Synthesis for Thin Shell Models. *Comput. Graph. Forum*, 2023, 42(5), doi:https://doi.org/10.1111/cgf.14903.

[39] Wei Y, Min S, Feng W, Zhu D, Mao T. Motion-Inspired Real-Time Garment Synthesis with Temporal-Consistency. *Journal of Computer Science and Technology*, 2023, 38: 1356–1368, doi:10.1007/s11390-022-1887-1.

[40] Feng W, Yu Y, Kim BU. A deformation transformer for real-time cloth animation. *ACM Trans. Graph.*, 2010, 29(4), doi: 10.1145/1778765.1778845.

[41] Bertiche H, Madadi M, Tylson E, Escalera S. DeePSD: Automatic deep skinning and pose space deformation for 3D garment animation. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, 5471–5480.

[42] Bertiche H, Madadi M, Escalera S. PBNS: Physically Based Neural Simulation for Unsupervised Garment Pose Space Deformation. *ACM Trans. Graph.*, 2021, 40(6), doi:10.1145/3478513.3480479.

[43] Liu L, Zheng Y, Tang D, Yuan Y, Fan C, Zhou K. NeuroSkinning: Automatic Skin Binding for Production Characters with Deep Graph Networks. *ACM Trans. Graph.*, 2019, 38(4), doi:10.1145/3306346.3322969.

[44] Li T, Shi R, Kanai T. DenseGATs: A Graph-Attention-Based Network for Nonlinear Character Deformation. In *Proc. Symp. Interactive 3D Graph. Games*, 2020, 5:1–5:9, doi:10.1145/3384382.3384525.

[45] Li T, Shi R, Kanai T. MultiResGNet: Approximating Nonlinear Deformation via Multi-Resolution Graphs. *Comput. Graph. Forum*, 2021, 40(2): 537–548, doi:https://doi.org/10.1111/cgf.142653.

[46] Xu Z, Zhou Y, Kalogerakis E, Landreth C, Singh K. RigNet: Neural Rigging for Articulated Characters. *ACM Trans. Graph.*, 2020, 39(4), doi:10.1145/3386569.3392379.

[47] Li P, Aberman K, Hanocka R, Liu L, Sorkine-Hornung O, Chen B. Learning Skeletal Articulations with Neural Blend Shapes. *ACM Trans. Graph.*, 2021, 40(4), doi:10.1145/3450626.3459852.

[48] Gundogdu E, Constantin V, Parashar S, Seifoddini A, Dang M, Salzmann M, Fua P. GarNet++: Improving Fast and Accurate Static 3D Cloth Draping by Curvature Loss. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, 44(01): 181–195, doi: 10.1109/TPAMI.2020.3010886.

[49] Li Y, Tang M, bo Yang Y, Huang Z, Tong R, Yang S, Li Y, Manocha D. N-Cloth: Predicting 3D Cloth Deformation with Mesh-Based Networks. *Comput. Graph. Forum*, 2022, 41(2): 547–558, doi:https://doi.org/10.1111/cgf.14493.

[50] Li YD, Tang M, Chen XR, Yang Y, Tong RF, An BL, Yang SC, Li Y, Kou QL. D-Cloth: Skinning-based Cloth Dynamic Prediction with a Three-stage Network. *Comput. Graph. Forum*, 2023, 42(7): 1–13, doi:https://doi.org/10.1111/cgf.14937.

[51] Shao Y, Loy C, Dai B. Towards Multi-Layered 3D Garments Animation. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, 14315–14324, doi:10.1109/ICCV51070.2023.01321.

[52] Yun S, Jeong M, Kim R, Kang J, Kim HJ. Graph Transformer Networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 2019, 11960–11970.

[53] Dwivedi VP, Bresson X. A Generalization of Transformer Networks to Graphs. *arXiv preprint*, 2020, arXiv: 2012.09699.

[54] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is All you Need. In *Proc. Adv. Neural Inf. Process. Syst.*, 2017, 5998–6008.

[55] Baraff D, Witkin A. Large Steps in Cloth Simulation. In *ACM SIGGRAPH*, 1998, 43–54, doi:10.1145/280814.280821.

[56] Atzmon M, Lipman Y. SAL: Sign Agnostic Learning of Shapes From Raw Data. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, 2562–2571, doi:10.1109/CVPR42600.2020.00264.

[57] He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet

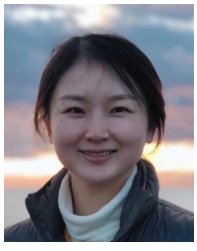
Classification. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, 1026–1034, doi:10.1109/ICCV.2015.123.

[58] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In *Proc. Int. Conf. Learn. Representations*, 2015, 1–15.

[59] Chen D, O'Bray L, Borgwardt KM. Structure-Aware Transformer for Graph Representation Learning. In *Proc. Int. Conf. Mach. Learn.*, 2022, 1–21.

[60] Yu YY, Choi J, Cho W, Lee K, Kim N, Chang K, Woo C, Kim I, Lee SW, Yang JY, Yoon SY, Park N. Learning Flexible Body Collision Dynamics with Hierarchical Contact Mesh Transformer. In *ArXiv*, 2023, 1–28.

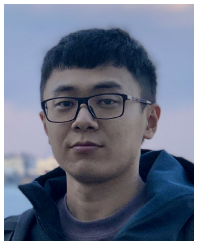
Author biography



Tianxing Li is a lecturer in the College of Computer Science, Beijing University of Technology. She received her Ph.D. degree in computer science from the University of Tokyo in 2021. Her current research interests include computer animation, visualization, and pattern recognition.

Zhi Qiao is a lecturer in China Agricultural University, Beijing. He received his Ph.D. degree in engineering from the University of Tokyo in 2021. His research interests center on computer graphics and computer vision.

Zihui Li is an undergraduate in the College of Computer Science, Beijing University of Technology. Her main interests are focused on 3D modeling, computer animation, and visualization.



Rui Shi is a lecturer in the School of Information Science and Technology, Beijing University of Technology. He served as a visiting scholar in the Department of General Systems Studies, University of Tokyo. He received his Ph.D. degree in computer science from the University of Tokyo in 2022. His current research interests include explainable artificial intelligence, computer animation, and visualization.

Qing Zhu is a professor in the College of Computer Science, Beijing University of Technology. She received her Ph.D. degree in electronic information and communication from Waseda University, Tokyo, Japan, in 2000. Her research interests include multimedia information processing technology, virtual reality technology, and information integration technology.