Spectrum-Enhanced Graph Attention Network for Garment Mesh Deformation

Tianxing Li, Rui Shi, Qing Zhu, Liguo Zhang, Takashi Kanai

Abstract—We present a novel solution for mesh-based deformation simulation from a spectral perspective. Unlike existing approaches that demand separate training for each garment or body type and often struggle to produce rich folds and lifelike dynamics, our method achieves the quality of physics-based simulations while maintaining superior efficiency within a unified model. The key to achieve this lies in the development of a spectrum-enhanced deformation network, a result of in-depth theoretical analysis bridging neural networks and garment deformations. This enhancement compels the network to focus on learning spectral information predominantly within the frequency band associated with intricate deformations. Furthermore, building upon standard blend skinning techniques, we introduce target-aware temporal skinning weights. The weights describe how the underlying human skeleton dynamically affects the mesh vertices according to the garment and body shape, as well as the motion state. We validate our method on various garments, bodies, and motions through extensive ablation studies. Finally, we conduct comparisons to confirm its superiority in generalization, deformation quality, and performance over several state-of-the-art methods.

Index Terms—Garment deformation, spectral bias, graph attention network, mesh-based simulation

1 Introduction

Cloth animation is a predominant domain in the computer graphics community that simulates the deformations of virtual garments resulting from temporal changes or external influences. Its applications span across various industries including film, video games, fashion design, virtual reality and augmented reality. Physics-based simulation (PBS) [1], [2] involves the creation of a dynamic model that leverages physical attributes of clothing, such as mass, elasticity, and friction, to mimic realistic garment deformation. However, the computational expense of PBS is substantial, presenting efficiency challenges when simulating large quantities of garments in practical applications. As the alternative solutions, deformation calculations can be simplified and expedited using techniques such as the skinning algorithms [3]–[5] or the pose space deformation methods [6], [7]. Nevertheless, the trade-off for these alternatives is often a compromise in the realism of the results obtained.

Leveraging deep learning models to predict garment deformation has emerged as a promising direction, offering a balance between efficiency and quality. Typically, learning-based methods [8]–[10] aim to train a nonlinear model using available data, enabling it to automatically generate garment mesh deformations based on input descriptions related to deformation. While these approaches excel in producing impressive results for tight-fitting garments, they encounter difficulties when it comes to generating dynamic behaviors for loose-fitting clothing such as dresses. More recently, some studies [11], [12] propose to use unsupervised learning strategies for deforming garments. These approaches offer an advantage of reducing the need for vast datasets and are capable of handling cloth dynamics by recasting the equations of motion as optimization problems, but they still lack the ability to produce detailed deformations on different garments within one model.

Crucially, we identify a central problem among existing learning-based approaches: the insufficient ability of networks to accurately learn intricate details of clothing deformations. Despite well-designed feature processing layers and network architectures can seemingly make reasonable induction, they still face the potential problem of spectral bias, *i.e.*, the middle and high frequency components where details are located are challenging to learn, leading to smoothness and stiffness of the results. This problem has been theoretically criticized in the machine learning community, but in the context of applications of graph neural networks (GNNs), the problem has not been explicitly discussed and resolved. Through the integration of theoretical analysis and experiments, we find that the key factor in addressing the challenge of accurately simulating intricate clothing deformations with neural networks lies in the reasonable enhancement of the spectral information represented by the networks.

Besides the quality of details, another major problem is the dynamic performance of loose-fitting garments. Existing deformation methods rely on the skinning weights of the underlying body or approximate garment weights through networks without fully considering the unique characteristics and motion state of the garment. This leads to models that lack robustness when dealing with more intense motions or garments exhibiting high variability, consequently

Tianxing Li and Qing Zhu are with the College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: litianxing@bjut.edu.cn; ccgsqz@bjut.edu.cn)
Rui Shi and Liguo Zhang are with the School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China (e-mail: ruishi@bjut.edu.cn; zhangliguo@bjut.edu.cn)
Takashi Kanai is with the Department of General Systems Studies, the University of Tokyo, Tokyo 153-8902, Japan (e-mail: kanait@acm.org)
Manuscript received 2 December 2023; revised 24 January 2025; accepted 1 May 2025. (Corresponding author: Rui Shi.)

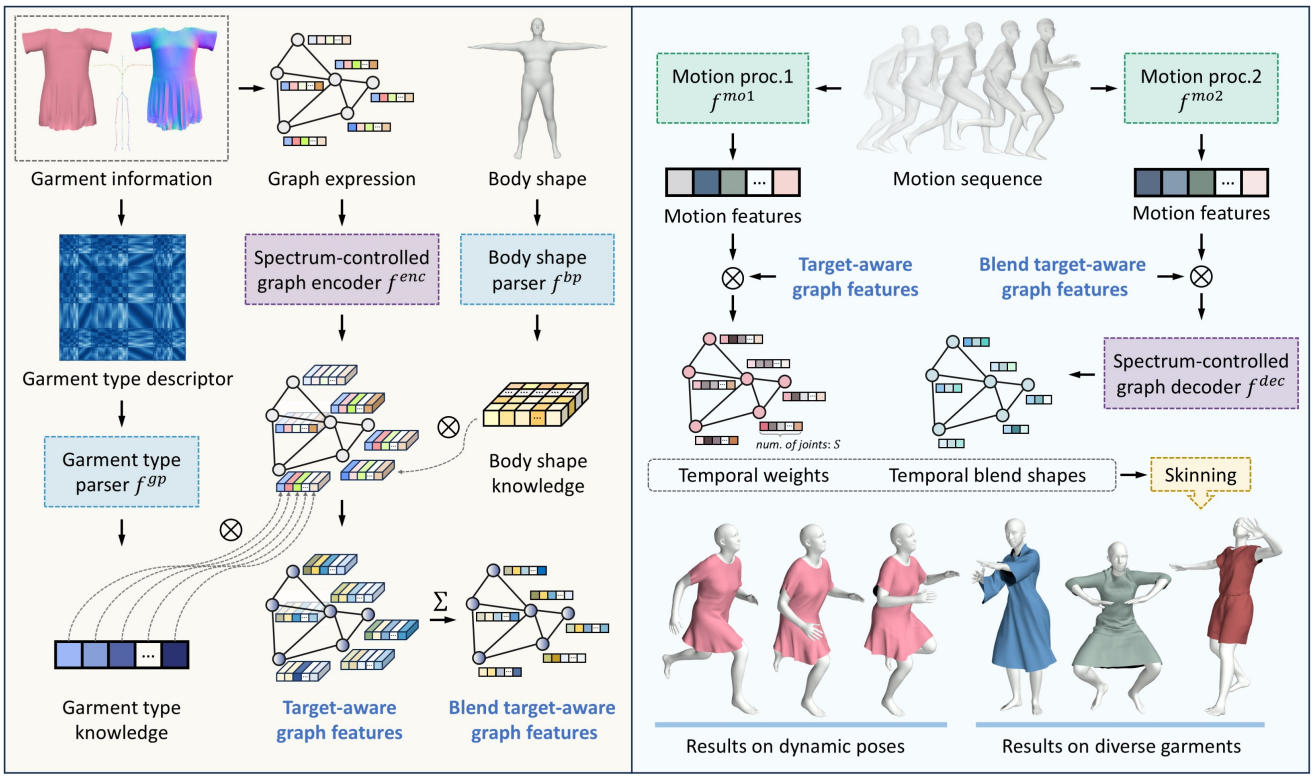


Fig. 1. The overview of our deformation network. The network operates as an end-to-end system, taking input in the form of garment information, body shape, and motion sequences to generate deformation results. On the left side of feature extraction, the graph encoder encodes rest-pose information and integrates it with the knowledge description of garment type and body shape, resulting in the target-aware graph features. Moving to the right side, deformation generation involves combining the graph features with motion data to derive temporal weights and blend shapes, ultimately leading to the deformed garment through skinning.

resulting in obvious artifacts.

In this work, we initiate with a theoretical analysis of spectral properties of neural networks and garment deformations. We then introduce a novel deformation method that effectively addresses the aforementioned problems. The proposal demonstrates advanced generalization abilities and dynamics of garments regardless of their type, mesh topology, vertex count, as well as underlying body shape and motion sequence. An overview of the proposal is shown in Fig. 1. The main contributions are as follows:

- **Spectrum-enhanced deformation network.** Based on the spectral decomposition attributes of garment mesh deformation, we propose a spectrum-enhanced graph attention (SEGA) block and formulate a unified deformation network built upon the spectrum enhancement of all parameterized hidden layers. Our approach effectively resolves the challenge of insufficient or unrealistic details resulting from spectral bias by constraining the network to learn spectral information predominantly within the frequency band associated with deformation details.
- **Target-aware temporal skinning weights.** Given the diversity of garments and bodies, along with the varying impact of motion states on different clothing types, we design target-aware temporal skinning weights that incorporate the garment type description, body shape, and motion. This fusion empowers the network to effectively handle the one-stage deformation generation of complex loose-fitting garments and diverse body shapes.

In terms of validation, we conduct comprehensive experiments to demonstrate the superiority of our proposal over state-of-the-art approaches in generalization, realism, and timing performance.

2 Related Work

In this section, we review existing methods for clothing animation, categorizing them into two main approaches: physics-based simulation and learning-based models, and then discuss the research and applications related to spectral bias.

Physics-based simulation. Physics-based methods are employed to achieve a high degree of realism in deformation effects. These methods build a model that considers the physical properties and dynamics of the garment. By numerically solving the model equations, the deformed mesh state at each time step is obtained, leading to realistic clothing behaviors [13]–[15]. In recent years, some research efforts have been devoted to developing methods [2], [16] for simulating hyperrealistic clothing, but obtaining impressive level of results comes at the expense of high computational costs. This computational expense remains a common challenge for physics-based simulation. To enhance efficiency, researchers have explored several approaches, including using position based dynamics [17]–[19], optimizing GPU-based algorithms [20]–[22], adding

fine details on coarse garment meshes [23]–[25], and trading realism and accuracy for better performance [26], [27]. Nonetheless, employing physics-based methods in real-time applications with limited computing resources remains an open challenge. Additionally, the task of setting appropriate simulation parameters is intricate and time-consuming. It often involves the laborious process of manually fine-tuning parameters to adjust cloth properties or mesh structure, requiring specific expertise. To address it, some approaches attempt to automatically extract physical parameters from an image [28], [29], a video [30], [31], or learn from data [32], [33]. In our method, we use physical simulation data as ground truth to train the neural network, ensuring efficient and high-quality deformation predictions. To define optimization objectives, we borrow some physics-based cloth simulators that can be formulated as time-varying partial differential equations, and use them as our loss functions, including gravity, stretch, and shear losses.

Learning-based models. With the advancements in deep learning, learning-based clothing deformation methods [34]–[39] have become widely adopted for their high efficiency and automation. Pioneering work [6] explores a pose space deformation which represents deformations as mappings from a pose parametric space. Building on this, several studies propose to learn garment deformations from pose, shape, or size parameters [40]–[43]. For generating garment deformations with style variations, Patel *et al.* [44] introduce TailorNet which regresses coarse and fine deformation in two steps via multi-layer perceptions (MLPs). However, the trained model has a large memory and can only handle garment meshes with specific topology.

To tackle the model generalization problem, researchers have turned to graph neural networks (GNNs), leveraging their powerful 3D data processing capabilities to handle arbitrary garment meshes. Liu *et al.* [45] work on applying graph convolution network to generate skinning weights for production characters with non-manifold meshes. Inspired by it, the subsequent studies [10], [46]–[53] have also utilized GNNs to predict skinning weights, blend shapes, or corrective displacement based on rough deformation for articulated characters. In a different line of research, researchers [8], [54] focus on the automatic generation of the detailed clothing effects. These methods typically leverage the SMPL parameterized human body [55] as a base, and the deformation of the garment is driven by the variations in the underlying body. Building upon PointNet [56], Gundogdu *et al.* [57] propose a two-stream architecture GarNet and introduce curvature to enhance clothing details. The approach yields results similar to physics-based simulations but achieves faster computation speed. Li *et al.* [58] propose SwinGar, a two-stage method that separates low- and high-frequency deformation generation. In the high-frequency part, they combine long short-term memory (LSTM) [59] and GNN to enhance garment details. Additionally, they design a frequency control technique to mitigate spectral bias during optimization. But their method only addresses graph convolutional layers and overlooks all other layers in their network. This oversight weakens the theoretical foundation of their approach, resulting in the need for a two-stage structure that significantly increases the running time and GPU memory usage for accurate predictions. In contrast, our method achieve to predict blend weights and blend shapes directly to generate deformation, offering faster execution and compatibility with all graphics engines.

In order to reduce the reliance on ground truth data, PBNS [11] proposes unsupervised solutions which formulating an implicit physics-based simulation by introducing several loss terms. While this method does not require data preparation, it suffers from repeated training when dealing with new garments. Subsequently, the authors present DeePSD [60], which utilizes a combination of supervised and unsupervised network structures to enhance deformation quality. Unlike PBNS, DeePSD enables the model to generalize to unseen garments without the need for additional training. However, both methods produce static deformations. To enable dynamic clothing effects, SNUG [61] incorporates an inertia term into the model but produces relatively stiff results with fold patterns. Similarly, Bertiche *et al.* [12] achieve cloth dynamics deformations using an unsupervised scheme, but the model is specific to the given body and garment, limiting its generalization. De Luigi *et al.* [62] model garments as unsigned distance fields to achieve the ability of processing diverse garment types, but their method primarily focuses on form-fitting garments in static scenarios. Grigorev *et al.* [63] employ a multiscale GNN to generate realistic dynamic deformations for arbitrary mesh topology. However, the designed pipeline is limited to one batch size, which affect its scalability and efficiency in real-world applications.

In an alternative line of research, approaches focus on clothing deformation from the perspective of computer vision, including texture learning [64]–[66], wrinkle style transfer [38], [67], and pixel-based clothing generation [68]–[71]. While these methods can achieve photo-realistic effects, they may struggle to capture 3D shape details accurately and can be sensitive to environmental factors like illumination and viewpoint.

Learning of different frequencies. Numerous prior studies on deep neural networks (DNNs) have consistently demonstrated a fundamental bias pattern in their learning process, *i.e.*, low-frequency functions are always learned first and better [72]–[75]. Consequently, the concept of reducing spectral bias to enhance neural network performance has garnered considerable attention within the machine learning community.

In a notable contribution, Rahaman *et al.* [76] link the Fourier spectrum of a neural network to the Lipschitz constant. They further explore the influence of the data manifold's shape, revealing that complex manifold shapes can facilitate the learning of higher frequencies. Related theoretical analysis and validations on different network structures can also be found in [77]–[81]. Regarding the capacity to learn higher-frequency information, an empirical evidence has established that deeper networks possess the capability to capture higher-frequency information that shallow networks cannot, as demonstrated in [82]–[84]. These studies demonstrate the theoretical feasibility of adjusting the learned spectra of DNNs.

In real-world applications, Tancik *et al.* [85] introduce a Fourier feature mapping technique to transform neural tangent kernels, achieving stationary states with adjustable bandwidth. Their work highlights the substantial impact of

frequency bands on 2D and 3D regression tasks. Miyato *et al.* [74] enhance the quality of image generation by fine-tuning the spectral features of hidden layers. Shi *et al.* [75] introduce a method for controlling spectral bias in convolutional layers, preventing eventual performance degradation, and expediting convergence in image inversion tasks. The use of spectral decomposition techniques to analyze consensus protocols within fixed topologies [86] and to represent the abstract traffic scenarios [87] also provides practical insights into these systems. These investigations not only validate the significance of adjusting frequencies in enhancing image generation and restoration, utilizing convolutional neural networks or attention-based transformers. They also directly inspire our idea of enriching deformation details of garment meshes through spectrum-enhanced GNNs.

3 Garment Neural Deformation

Given an initial garment mesh template $\mathbf{T}$ with N vertices, an SMPL [55] human body with shape descriptor β, and a sequence of T pose transformation information $\Theta = \{\theta_t, \theta_{t-1}, \ldots, \theta_{t-T+1}\}$ controlled by corresponding skeleton $\mathcal{J}$, our approach aims to learn a deep neural network $\mathcal{F}$ for predicting clothing deformation on top of the animated human body. The overview of our proposal can be formulated as follows:

$$\mathcal{M}^t = \mathcal{F}(\mathbf{T}, \beta, \mathcal{J}, \Theta), \tag{1}$$

where $\mathcal{M}^t$ is the deformed garment mesh at current state t. Intuitively, $\mathcal{F}$ should be designed to fulfil the following requirements: a) the capability to process diverse garment meshes and effectively represent their personalized behaviors, b) the ability to handle time series of body movements, and c) the realization of intricate clothing deformations with particular attention to details. In this work, instead of directly yielding the vertex positions of the garments $\mathcal{M}^t$ in the animation, we select to force the network to generate the temporal skinning weights $\widetilde{\mathcal{W}}$ and the blend shapes B, which are used for deforming $\mathbf{T}$ by a skinning algorithm $W(\cdot)$ (*e.g.*, linear blend skinning or dual quaternion):

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

$$\widetilde{\mathcal{M}}(\beta, \Theta, \mathbf{T}) = \mathbf{T} + B(\beta, \Theta, \mathbf{T}). \tag{3}$$

In particular, as shown in Fig. 1, our end-to-end deformation network $\mathcal{F}$ primarily consists of six modules. More specifically, we rely on a garment type parser f^{gp}, a body shape parser f^{bp}, a garment graph encoder f^{enc}, and a motion processor f^{mo1} to compute the garment skinning weights $\widetilde{\mathcal{W}}$ so that the produced weights are garment-, body-, as well as motion-dependent. Additionally, we fuse the high-level garment features from the encoder with features from the two parsers and another motion processor f^{mo2}, then, forward the result into the garment decoder f^{dec} to estimate blend shapes B. In its entirety, the workflow composed of six modules can be summarized into two parts: feature extraction (left half of Fig. 1) and deformation generation (right half of Fig. 1).

To achieve the above regression, we employ linear layers and non-linear activations for feature extraction in the parsers of f^{gp} and f^{bp}, LSTM for time-varying feature transformation in the motion processors f^{mo1} and f^{mo2}, and graph learning layers to capture graph-structured information in the garment encoder f^{enc} and decoder f^{dec}. Among the various types of information processing, the most intricate part involves extracting and propagating garment graph features, leading to vertex-level predictions, which we elaborate in the subsequent section.

4 Methodology

4.1 Graph Learning for Garments

Handling information from garment meshes presents a challenging task due to their arbitrary topologies and vertex counts. A promising solution is to represent the mesh as a graph and apply graph learning algorithms, which has gained attention in garment animation field in recent years. In our work, we represent a garment mesh $\mathbf{T}$ as a graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X} = \{x_1, \ldots x_N\}$ represents the set of features from N vertices and $\mathcal{E}$ denotes the connectivity between vertices. For each vertex, the features are defined as: $x_i = [p_i^\top, n_i^\top, d_i^\top] \in \mathbb{R}^F$, which consists of the position $p_i^\top \in \mathbb{R}^3$, the normal $n_i^\top \in \mathbb{R}^3$, and the distance to all skeletal joints of the body $d_i^\top \in \mathbb{R}^S$, where S denotes the number of the SMPL body joints. This feature assignment makes the garment graph highly expressive, providing a comprehensive representation of both the local garment information and the positional relationships between vertices and the driving skeleton in rest pose.

Given the constructed garment mesh graph, the next step is to process it with graph learning techniques to effectively extract high-level representations. Graph attention networks [88] leverage the attention mechanism to concentrate on significant neighbors of each vertex in the graph, allowing the network to identify and propagate meaningful information to subsequent steps. The graph attention block (GATB) has been verified for aggregating complex graph information and handling nonlinear deformation tasks [48], [89]. The processing of the graph feature matrix $X = [x_1, \ldots x_N] \in \mathbb{R}^{F \times N}$ can be expressed as follows:

$$\Psi(X) = v(X)\sigma(s(X))^\top \| m(X), \tag{4}$$

where $s : \mathbb{R}^{F \times N} \to \mathbb{R}^{N \times N}$ is for computing the attention scores, and σ represents the softmax function for normalization. The attention score matrix $s(X)$ is computed dynamically based on pairwise relationships between vertices. Its calculation is local and vertex-wise, relying only on the features of individual vertex pairs (x_i and x_j), rather than the total number of vertices N. The softmax normalization further ensures scalability by acting independently for each vertex, allowing the GATB to handle graphs with varying N without requiring resampling or a fixed graph size. For a more detailed discussion of the graph attention mechanism, we recommend [88] as a reference. $v, m : \mathbb{R}^{F \times N} \to \mathbb{R}^{F' \times N}, \mathbb{R}^{F'' \times N}$ are the linear transformations. F, F', and F'' stand for feature dimensions of corresponding submodules. The symbol $\|$ denotes the concatenation, concatenating the attention weighted part $\Phi(X) = v(X)\sigma(s(X))^\top$ and the self-reinforcement stream $m(X)$. In this way, features can be

efficiently transmitted to deeper layers through attention aggregation and self-reinforcement, enabling the computation of hidden representations of irregular mesh graphs.

4.2 Spectrum-Enhanced Graph Learning

Frequency of deformation details. From a spectral point of view, garment deformation is commonly divided into a low-frequency part, representing overall changes that follow the body's movements, and a high-frequency part, which captures intricate details such as wrinkles. Building upon this notion, several studies [10], [44], [63] have adopted the stepwise deformation learning framework to address the complex task of deformation prediction. However, these methods have not fully overcome the difficulty of insufficient garment details, particularly when dealing with multiple garment types with a single neural network. The underlying reason for such limitations lies in the difficulties of deep neural networks to effectively process the required frequency band information. A similar spectral bias phenomenon has been observed in image prior, with some basic ideas for measuring and controlling frequency information discussed in [75]. However, the ability of GNNs to learn different frequency information is still poorly explored.

To solve this problem fundamentally, we first perform a harmonic analysis on the frequency components of the garment mesh. Our crucial observation is that the observable deformation details are not confined to the highest frequencies, but instead, they predominantly exist in the middle and relatively high frequency ranges. To effectively capture these different frequency components, we perform a decomposition of a garment's Laplacian matrix and sort the corresponding eigenvectors based on their eigenvalues. Specifically, we select about 3% eigenvectors corresponding to the smallest eigenvalues for reconstructing the low-frequency part, 3% to 30% eigenvectors for reconstructing the middle frequency part, while the remaining eigenvectors are used to represent the highest frequency part [90]. It is noteworthy that, due to the absence of an explicit definition, this division into low, middle, and high frequencies are based on empirical observations relying on exponential correlations between different frequency components. As shown in Fig. 2, the low frequencies primarily capture rough, smooth deformations, whereas the mid-range frequencies contain nearly all deformation details. Notably, the high-frequency details in the highest frequency band are almost imperceptible. This example highlights the importance of a network's ability to accurately learn information in the middle and higher frequency bands to achieve successful garment deformation. Therefore, our main proposal lies in designing a spectrum-enhanced strategy for controlling detail learning, which will be further elaborated in the following.

Basics of spectrum-enhancement. Adjusting the network's Fourier spectrum to limit the fitting of unimportant frequencies is a feasible approach to prevent model performance degradation during mid-frequency deformation learning and improve the learning of core details. With this basic idea, we aim to upper bound the Fourier coefficients of the core component (*i.e.*, GATB) in our network, for the sake of constraining the Fourier spectrum of the network. Following the principles of Lipschitz continuity as described

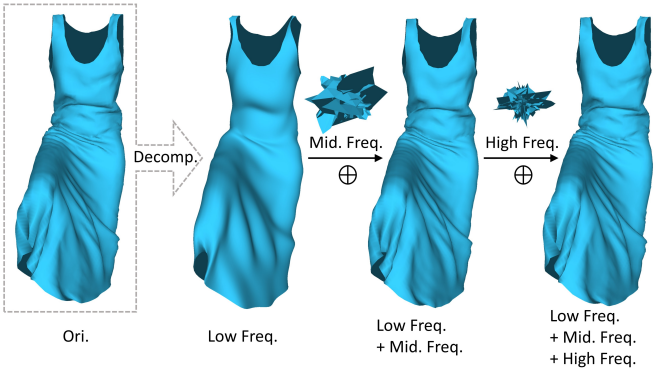


Fig. 2. Decomposition of different frequency components of an example garment. The "Low frequency + Mid. Frequency" part visually resembles the original garment closely, while the high(est)-frequency part is almost imperceptible to the naked eyes.

in [75], [80], [81], we can achieve this by enforcing Lipschitz continuity on GATB. Specifically, a Lipschitz continuous function is derivable almost everywhere [91], *i.e.*, derivable at every data point outside a set of Lebesgue measure zero, and the Lipschitz constant L_F of the function is:

$$L_F(\Psi) = \sup_{X \in F} |||\mathbf{D}\Psi_X|||_F, \tag{5}$$

where $|||\Psi|||_F = \max_X \|\Psi(X)\|_F / \|X\|_F$ and $\|\cdot\|_F$ is Frobenius norm. $\mathbf{D}\Psi_X$ stands for the Fréchet derivative of GATB at X. According to harmonic analysis theory [92], the Fourier coefficients of Ψ are bounded by the Lipschitz constant:

$$|c_k| \leq \frac{L_F(\Psi)}{Ck^2}, \tag{6}$$

where c_k denotes the k-th Fourier coefficient, $|\cdot|$ stands for the norm of a complex number. C is a constant during deriving the coefficient bound. We can find that the key to determining the upper bound on the Fourier coefficients lies in adjusting $L_F(\Psi)$. However, in the absence of any constraints over the attention layer, $L_F(\Psi)$ remains uncertain.

Next, we present theorems that establish the relationship between the upper bound of the Lipschitz constant and GATB. Additionally, we introduce a scaling strategy to effectively enforce spectral adjustment.

Theorem 1. *Given a GATB function, if both the attention weighted part and the self-reinforcement stream are Lipschitz continuous, then the GATB as defined in Eq.* (4) *is Lipschitz continuous and*

$$L_F(\Psi) \leq \sqrt{L_F(\Phi)^2 + L_F(m)^2}. \tag{7}$$

A detailed proof and discussion of the theorem and subsequent lemmas can be found in supplementary materials. As shown in Eq. (7), if we can determine L_F for both parts, we can obtain the desired upper bound for GATB. In line with previous studies that constrain frequency information or Lipschitz constant within various network structures [74], [78]–[81], we also select to reconstruct the hidden layers to attain the Lipschitz constant's upper bound, which in turn bounds the Fourier coefficients. On the one hand, m is a linear function, and its Lipschitz constant $L_F(m)$ is equal to the spectral norm of its parameters $\|W^m\|_*$, where W^m represents the parameters of the operation m. Therefore,

$L_F(m)$ can be upper bounded by constraining its spectral norm to a constant value τ using a scaling function. The linear function $m(X)$ is then formulated as:

$$m(X) = \frac{\tau W^m X}{\max(\tau, \|W^m\|_*)}. \tag{8}$$

On the other hand, $L_F(\Phi)$ is still unknown. Next, we will demonstrate how to determine its upper bound, which is also the upper bound of $\|\mathbf{D}\Phi_X\|_F$.

Norm of attention derivative. For the input feature $X \in \mathbb{R}^{F\times N}$, the norm of the Fréchet derivative is upper bounded by:

$$\begin{aligned}\|\mathbf{D}\Phi_X\|_F &\le \left\|\mathbf{D}v_X(H)\sigma(s(X))^\top\right\|_F \\ &\quad + \left\|v(X)\mathbf{D}\sigma_{s(X)}(\mathbf{D}s_X(H))^\top\right\|_F \qquad (9a)\\ &\le \|\mathbf{D}v_X\|_F \|\sigma(s(X))\|_F \\ &\quad + \sqrt{2}\left\|v(X)^\top\right\|_{(\infty,2)} \|\mathbf{D}s_X\|_F, \qquad (9b)\end{aligned}$$

where $v(X)^\top$ is a matrix (temporarily denoted as M for simplicity), and $\|M\|_{(\infty,2)} = \max_i(\sum_j M_{i,j}^2)^{1/2}$. $\mathbf{D}s_X$ can be regarded as a function (temporarily denoted as f), and $\|f\|_F = \max_X \|f(X)\|_F / \|X\|_F$ (resp. $\|f\|_{F,1}$ in Eq. (11)). Eq. (9b) shows that $L_F(\Phi)$ is determined by two terms: the first one is related to the output linear operation and the uniformity of the probabilities, while the second one is related to the range of the output linear operation and the gradient of the score function.

Next, we will examine these two terms in Eq. (9b) through **Lemma 1** and **2**, and demonstrate that bounding the spectral norm of the linear operation and the score function enables us to control both terms simultaneously.

Lemma 1. *If the spectral norm of the weights of the linear operation $v(X)$ is bounded by τ and the value of the score function $s(X)$ is bounded by α, then the first term of Eq.* (9b) *is upper bounded by*

$$\|\mathbf{D}v_X\|_F \|\sigma(s(X))\|_F \le \tau e^\alpha, \tag{10}$$

where α and τ are constants used as bounds for the function. The proof of **Lemma 1** can be found in supplementary materials.

For the second term of Eq. (9b), if there are no additional constraints, it cannot achieve a constant upper bound because the output linear operation $\left\|v(X)^\top\right\|_{(\infty,2)}$ changes with the input and parameters. To address this issue, we introduce another scaling function $n : \mathbb{R}^{F\times N} \to \mathbb{R}^+$ into the original score function to achieve a tight bound of the second term. The score function is scaled as $s(X) = s^{pre}(X)/n(X)$, where $s^{pre}(X)$ is the original unscaled score function. Then, we have:

$$\begin{aligned}&\|v(X)^\top\|_{(\infty,2)}\|\mathbf{D}s_X\|_F \le \\ &\qquad \frac{\left\|v(X)^\top\right\|_{(\infty,2)} \|\mathbf{D}s_X^{pre}\|_F}{n(X)} + \\ &\qquad \frac{\left\|v(X)^\top\right\|_{(\infty,2)} \|\mathbf{D}n_X\|_{F,1} \|s^{pre}(X)\|_F}{n(X)^2}.\end{aligned} \tag{11}$$

Lemma 2. *With a wise choice of the scaling function n, the right side of Eq.* (11) *can be constrained to be a constant, i.e.*

$$\left\|v(X)^\top\right\|_{(\infty,2)} \|\mathbf{D}s_X\|_F \le \alpha + \alpha\tau. \tag{12}$$

Choice of scaling function. When the score function s is Lipschitz, **Lemma 2** can be satisfied by setting the scaling function as follows

$$n(X) = \frac{\max\{\|s^{pre}(X)\|_F, \left\|v(X)^\top\right\|_{(\infty,2)} L_F(s^{pre})\}}{\alpha}, \tag{13}$$

where the denominator ensures that the gradient of the scaled scores remains low compared to the input features. Further discussion can be found in supplementary materials.

Theorem 2. *If both $\Phi(X)$ and $m(X)$ are constrained, the Lipschitz constant of GATB is upper bounded by*

$$L_F(\Psi) \le \sqrt{\left(\tau e^\alpha + \sqrt{2}(\alpha + \alpha\tau)\right)^2 + \tau^2}. \tag{14}$$

By scaling the linear operations and the score function within our network to adjust the upper bound of the Fourier coefficients of hidden layers (*i.e.*, linear and graph attention layers), we can constrain the spectrum range that the network learns, thereby directing its attention towards the desired frequency bands for garment deformation (*i.e.*, the middle and higher frequencies). In essence, the parameters α and τ act as hyperparameters of the GATB, directly controlling its spectral expressiveness. The modified GATB, referred to as spectrum-enhanced graph attention block (SEGA), serves as the basis for our network. It is noteworthy that our SEGA does not require the transformation of input features into the frequency domain. By tuning the parameters α and τ, we can directly influence the frequency range that the network targets, encouraging a focus on the mid-frequency band. This spectrum-enhanced approach leads to more accurate and controllable modeling of garment deformation.

4.3 Target-Aware Temporal Weights and Blend Shapes

Once the network with SEGA basis is established, our next objective is to generate the skinning weights and blend shapes for garments deformation.

Skinning weights indicate the extent to which each bone in the skeleton affects a vertex, and we observed that various types of garments should be assigned vertex weights with varying degrees of bias. For example, the lower body bones should exert a stronger influence on the hemline vertices of a dress compared to the leg vertices of a trouser suit, so that the corresponding garment can smoothly follow movements and produce a natural swinging effect. However, existing learning-based deformation methods either assign constant weights to garments based on nearest-neighbor body points in the rest pose [41], [44], [61], or optimize the weights along with the network parameters but do not fully consider the garment object characteristics and the state of movement in which they deform [11], [60]. These implementations fall short in realistically modeling the intricate dynamics of personalized clothing, especially for loose-fitting garments that do not tightly conform to the body. Therefore, it is essential to first explore a garment global shape representation method for diverse garments and then integrate it into the skinning weight generation.

Garment type descriptor. Inspired by mesh spectral analysis, we find that Laplacian eigenvalues and can reveal

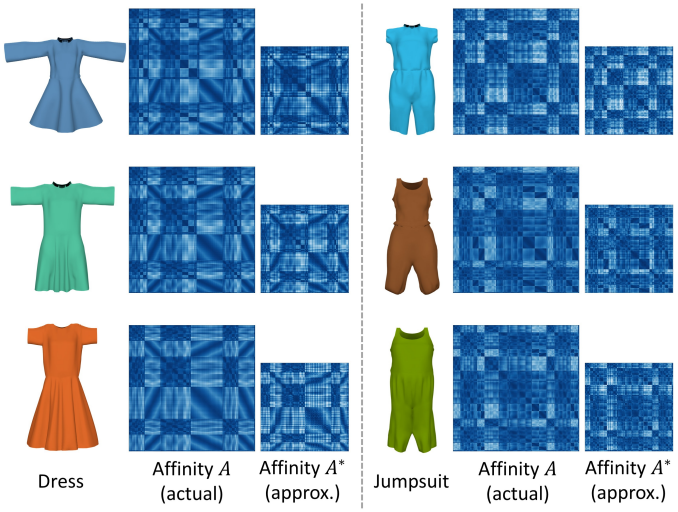


Fig. 3. Affinity matrices A and A^* for various dresses and jumpsuits. The size N^* of approximated A^* is fixed through all garments. The larger the value in the matrix, the darker the blue color. Similarities are observed among matrices of the same garment type, with noticeable differences attributed to the length of sleeves or pants legs.

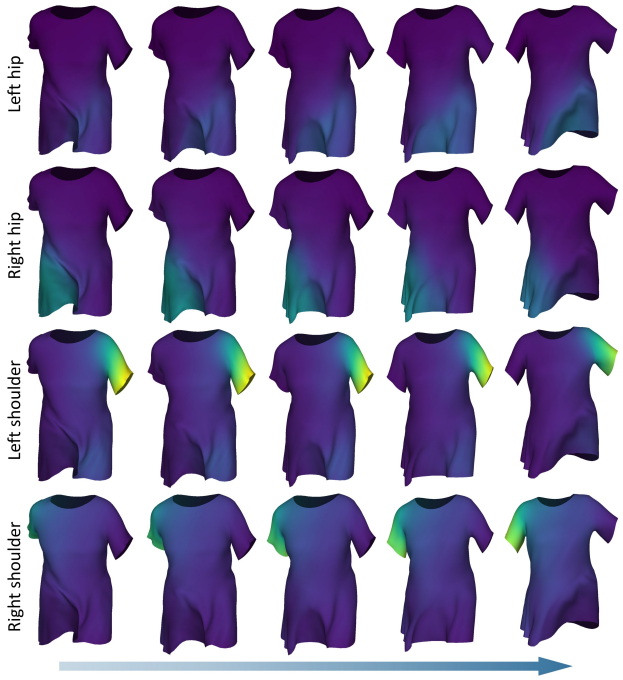


Fig. 4. Temporal skinning weights. The weights of joints affecting garment vertices change dynamically during the motion sequence, facilitating reasonable global deformations.

the intrinsic characterization of the shape structure. Specifically, starting from a garment mesh with N vertices, we build an affinity matrix $A \in \mathbb{R}^{N\times N}$ using a Gaussian function, where $A_{i,j} = \exp(-d_{i,j}^2/2r^2)$ with $d_{i,j}$ representing the geodesic distance between vertex i and j, and r being Gaussian kernel width set to the maximum geodesic distance between any two vertices. Due to the time complexity for eigen decomposition of an $N \times N$ affinity matrix, we adopt the Nyström approximation [93] to efficiently approximate the leading eigenvalues and eigenvectors. To achieve this, we begin to employ the furthest point sampling which involves randomly selecting an initial vertex and iteratively adding the vertex farthest from the already-sampled vertices until a fixed number N^* of vertices is reached, where $N^* \ll N$. Once the set of sampled vertices is obtained, we construct its smaller affinity matrix $A^* \in \mathbb{R}^{N^*\times N^*}$, which is then decomposed as $A^* = U\Lambda U^\top$. Fig. 3 illustrates the actual affinity matrix A and the sampled smaller affinity matrix A^* for dresses and jumpsuits. It can be seen that the matrices can provide effective shape representations and also tend to be global in nature. Consequently, we combine the eigenvalues from A^* into a vector $\lambda = [\lambda_1, ..., \lambda_{N^*}]$, which serve as our garment global shape descriptor. This descriptor enables differentiation between various types of garments, thereby enhancing the network's ability to generalize across different styles. Theoretically, it allows for generalization to garments with any number of vertices, not limited to those within the range encountered in the training garments.

Target-aware temporal weights. Recent studies have revealed that using fixed skinning weights [41], [44] for garments fails to accurately represent phenomena such as the sliding of loose-fitting garments on the body during movement. To improve this, dynamic skinning weights [35], [38] have been proposed, calculated by averaging the body vertices close to the garment in each frame. This method helps mitigate the unnatural rigidity often observed with static weights, yet several challenges persist. Firstly, the skinning weights [35] only consider the current state, overlooking crucial temporal sequential information that significantly influences garment dynamics. Secondly, they [35], [38] fail to consider the response properties of the target (*i.e.*, garment or body) in the skinning weights. Integrating this information is essential for developing a generalized deformation prediction model.

To overcome these shortcomings, our method enhances the generation of garment skinning weights by explicitly incorporating prior knowledge from the garment, body, and motion information into the graph embedding of each vertex. Concerning the garment features, as depicted in the left part of Fig. 1, we initially process the garment type descriptor λ using a type parser f^{gp} to extract garment type knowledge. Additionally, the 3D garment mesh graph $\mathcal{G}$ is encoded using a spectrum-enhanced graph encoder f^{bp}. For the body object, we handle the shape parameter β from the SMPL human model through a body shape parser to derive body shape knowledge. These three processed features are then integrated to create the target-aware graph features:

$$P_{i,j,k} = \sum_{i'=1}^{F^{enc}} f^{gp}(\lambda)_{i'} f^{enc}(\mathcal{G})_{i',j,k} f^{bp}(\beta)_{i,j}, \qquad (15)$$

where $f^{gp} : \mathbb{R}^{N^*} \to \mathbb{R}^{F^{enc}}$, N^* is a fixed number representing the length of eigenvalues of A^*, F^{enc} stands for the output feature dimension of the spectrum-enhanced graph encoder. $f^{bp} : \mathbb{R}^{|\beta|} \to \mathbb{R}^{F^{enc}\times S}$, $|\beta|$ is the length of the SMPL body shape descriptor, S is the number of body joints, also corresponding to the head number of graph attention operation. $f^{enc} : \mathbb{R}^{F\times N} \to \mathbb{R}^{F^{enc}\times S\times N}$, F means the input feature dimension.

In addition to target-awareness, it is essential for the skinning weights to adaptively respond to changes in body movement. To achieve this, we utilize a motion parser f^{mo1} to process the consecutive movement $\Theta = \{\theta_t, \theta_{t-1}, \ldots, \theta_{t-T+1}\}$ from SMPL human body poses $\theta \in \mathbb{R}^{3(S+1)}$ into high-level features. These features are then infused with the target-aware graph features to compute the adaptive skinning weights:

$$\widetilde{\mathcal{W}}_{j,k} = \sum_{i=1}^{F^{enc}} P_{i,j,k} f^{mo1}(\Theta)_i, \tag{16}$$

where $f^{mo1} : \mathbb{R}^{T\times 3(S+1)} \rightarrow \mathbb{R}^{F^{enc}}$. T is the number of frames included in dynamic information. This multi-feature integration allows personalized features to directly influence the skinning weights, providing the network with expressive power to handle diverse data. An example of generated target-aware temporal skinning weights is depicted in Fig. 4, demonstrating how they dynamically adjust in response to the motion state.

Target-aware temporal blend shapes. To compute blend shapes B, we also combine the processed features from two parsers and the graph encoder to generate the blend graph features that integrate body and garment awareness. Given that the deformation information contained in blend shapes is considerably more complex than skinning weights, we introduce an additional decoder to generate detailed corrections. In particular, the features are forwarded into the garment decoder to predict the blend shape:

$$X^{mid}_{i,k} = P'_{i,k} f^{mo2}(\Theta, \beta)_i, \tag{17}$$

$$B = f^{dec}\left(\mathcal{G}^{mid}\right), \tag{18}$$

where $f^{mo2} : (\mathbb{R}^{T\times 3(S+1)}, \mathbb{R}^{|\beta|}) \rightarrow \mathbb{R}^{F^{enc}S}$ and the feature channels of P is flattened and concatenated to produce P' for the element-wise multiplication. X^{mid} represents the middle feature before feeding into the spectrum-enhanced graph decoder f^{dec}. $f^{dec} : \mathbb{R}^{(F^{enc}S)\times N} \rightarrow \mathbb{R}^{3\times N}$. The graph $\mathcal{G}^{mid}$ is composed by the vertex feature X^{mid} and the vertex connectivity $\mathcal{E}$.

After calculating $\widetilde{\mathcal{W}}$ and B, the deformed garment mesh can be computed with Eqs. (2) and (3). Instead of predicting vertex positions, our methodology predicts blend skinning weights and blend shapes. This pipeline aligns with the industry standard in 3D animation, ensuring the compatibility with all major graphics engines. Our one-stage method also increases computational efficiency when compared to the two-stage methods such as TailorNet [44] and SwinGar [58], because it eliminates the operations of intermediate data conversion and independent high-frequency detail generation.

5 Experiments

Our neural network consists of six functionally distinct modules, and we use both supervised and unsupervised losses for training. The dataset includes various types of garments from Cloth3D [94], human bodies from SMPL [55], and motion sequences from the CMU Mocap [95]. Further details on network parameterization, training implementation, and dataset description are provided in the supplementary material.

5.1 Generalization Assessment

In our study, we represent garments as graphs, enabling the handling of garments with diverse topologies. To evaluate its generalzation capability of our model, we conduct tests on unseen garments and display the deformation results in Fig. 5. For each test garment, we also present similar training garments with gray color to provide a context for comparison. We measure dissimilarity between a test object $\mathbf{T}^P$ and a training object $\mathbf{T}^Q$ using the chi-square distance applied to their respective global shape descriptors, as defined in Sec. 4.3. This dissimilarity metric quantifies how different the test and training garments are in terms of their shapes:

$$\text{Dissim}(\mathbf{T}^P, \mathbf{T}^Q) = \frac{1}{2}\sum_{i=1}^{N^*} \frac{(|\lambda_i^{\mathbf{T}^P}|^{\frac{1}{2}} - |\lambda_i^{\mathbf{T}^Q}|^{\frac{1}{2}})^2}{|\lambda_i^{\mathbf{T}^P}|^{\frac{1}{2}} + |\lambda_i^{\mathbf{T}^Q}|^{\frac{1}{2}}}. \tag{19}$$

For each test garment, we present the four training garments that exhibit the least dissimilarity, highlighting noticeable distinctions in various aspects such as sleeve length, hemlines, and trouser legs.

More specifically, among the six test garments, the three depicted in the upper part of Fig. 5 exhibit similarities to the training garments and basically fall within the training data distribution. In contrast, the lower three garments (T-shirt, vest, and cardigan) are completely different in type from those used in training, such as dresses and jumpsuits. These garments lie outside the training data distribution, as illustrated by the distribution visualization in the supplementary material. Despite these deviations, our method successfully generates plausible deformations with rich details. Here, the prediction of deformations for the cardigan is particularly challenging, given the absence of similar front-open type garments in our dataset. This type of garment tends to fit more closely around the front opening, sometimes making it difficult to achieve a natural draping effect. Nonetheless, our model is still capable of accurately reproducing the intricate details of the folds in the cardigan's back area. This experiment shows that our method can generate realistic deformations for unseen garments with diverse garment types. It remains effective even when the test data deviates from the training sample distribution.

In addition to its versatility with different garments, our model also has the ability to generalize across diverse body shapes. As depicted in Fig. 6, three different body shapes are wearing the same dress and performing the same dance motion sequence. By incorporating body shape knowledge into our graph features, our model can predict tailored folds and wrinkles for various body shapes.

5.2 Evaluation on Spectrum Enhancement

Effect of SEGA parameters on overall deformation. To highlight the impact of our proposed spectrum-enhanced network, we conduct a comparative analysis of the network's performance under various settings of boundary parameters α and τ. Table 1 presents a quantitative result of the disparities between our predictions and physics-based simulations, considering vertex positions, vertex normal angles, and edge lengths. All test garment types can be found in supplementary materials. Here, we empirically

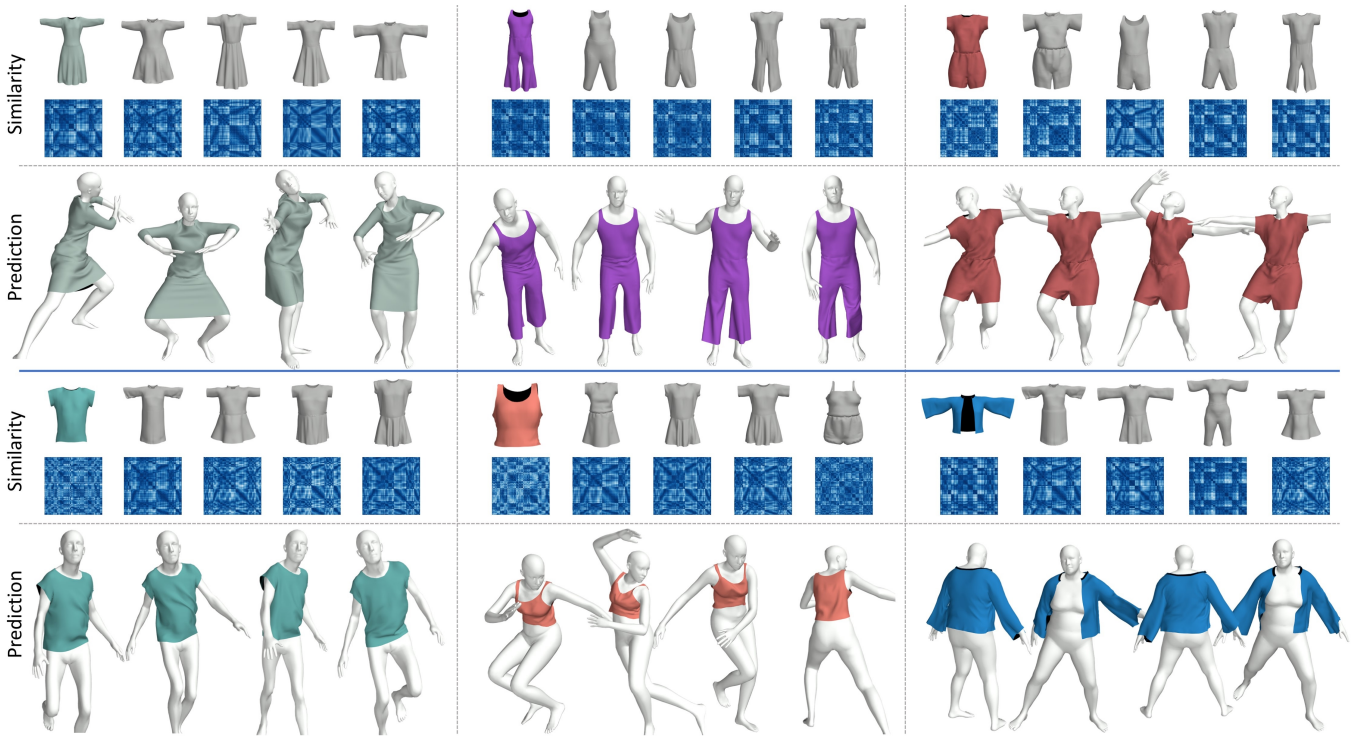


Fig. 5. Qualitative results on test data. We emphasize our generalization ability with diverse clothing by visualizing the dissimilarity between test and training garments: the four training clothing items (in gray) that exhibit the closest global shapes to each test case (in color), along with their corresponding affinity matrices.

TABLE 1
Boundary parameter choices for network spectrum-enhancement. The values before and after the slash symbol indicate the average vertex distance (cm), average angular deviation (°) of vertex normals, and average edge length (mm) between predictions and PBS. Setting the hyperparameters α and τ to values of 2 achieves optimal performance across most evaluated metrics.

	Long Dress	Short Dress	Long Jumpsuit	Short Jumpsuit
w/o	3.26 / 18.9 / 0.61	2.97 / 16.5 / 0.48	2.86 / 15.7 / 0.49	2.91 / 16.4 / 0.44
$\alpha = 1, \tau = 1$	2.61 / 16.7 / 0.48	2.56 / 15.5 / 0.35	2.35 / 12.2 / 0.37	2.31 / 13.6 / 0.34
$\alpha = 1, \tau = 2$	2.58 / 15.8 / 0.45	2.53 / 15.1 / 0.35	2.32 / 11.8 / 0.36	2.28 / 13.5 / 0.34
$\alpha = 1, \tau = 3$	2.59 / 17.3 / 0.50	2.52 / 15.4 / 0.35	2.31 / 10.7 / 0.36	2.22 / 12.7 / 0.29
$\alpha = 2, \tau = 1$	2.53 / 15.0 / 0.44	2.48 / 14.6 / 0.33	2.27 / 11.4 / 0.34	2.23 / 13.1 / 0.31
$\alpha = 2, \tau = 3$	2.51 / 14.4 / 0.33	2.42 / 13.8 / 0.29	2.21 / **10.4** / **0.30**	2.17 / 12.1 / 0.26
$\alpha = 3, \tau = 1$	2.56 / 15.4 / 0.44	2.51 / 14.9 / 0.34	2.29 / 11.4 / 0.34	2.25 / 13.2 / 0.32
$\alpha = 3, \tau = 2$	2.55 / 15.6 / 0.44	2.51 / 15.4 / 0.34	2.29 / 10.9 / 0.35	2.25 / 12.9 / 0.31
$\alpha = 3, \tau = 3$	2.49 / 14.7 / 0.33	2.47 / 14.2 / 0.29	2.25 / 11.0 / 0.32	2.21 / 12.5 / 0.27
$\alpha = 2, \tau = 2$	**2.46** / **14.2** / **0.42**	**2.37** / **13.3** / **0.27**	**2.20** / 10.5 / **0.30**	**2.13** / **11.7** / **0.25**

examine three values for each parameter. The variations in error across different settings are relatively minor, with the best performance observed for $\alpha = 2, \tau = 2$, and the worst for $\alpha = 1, \tau = 1$. The parameters α and τ can take decimal values. However, considering the quality of the experimental results, it is not cost-effective to spend computational resources on finding the ultimate parameter-optimized solution; hence, we do not conduct further experiments with higher precision. It is noteworthy that we also evaluate the network without spectrum-enhancement, which yielded obviously higher errors. This result stands in stark contrast to the better performance of the proposed SEGA.

To further validate the effectiveness of the spectrum enhancement strategy and the impact of its parameter settings during training, we select the top two performing configurations and the lowest-performing configuration from Table 1. We conduct an ablation study by plotting the average vertex errors on the validation data during the course of network training. In Fig. 7, the orange line represents the network in its original state without spectrum enhancement, while other lines, depicted in various colors, represent the network's performance when SEGA is applied with different parameter settings. Theoretically, the hyperparameters α and τ impact the model's hidden layers in distinct ways. Specifically, τ primarily affects the spectrum range of the linear layer, while α primarily influences the attention computation. Together, these parameters facilitate spectral control within the SEGA framework. If excessively large values are chosen for either parameter, the constraints that the method is intended to impose become ineffective, leading to a loss of meaningful control and causing instability in the

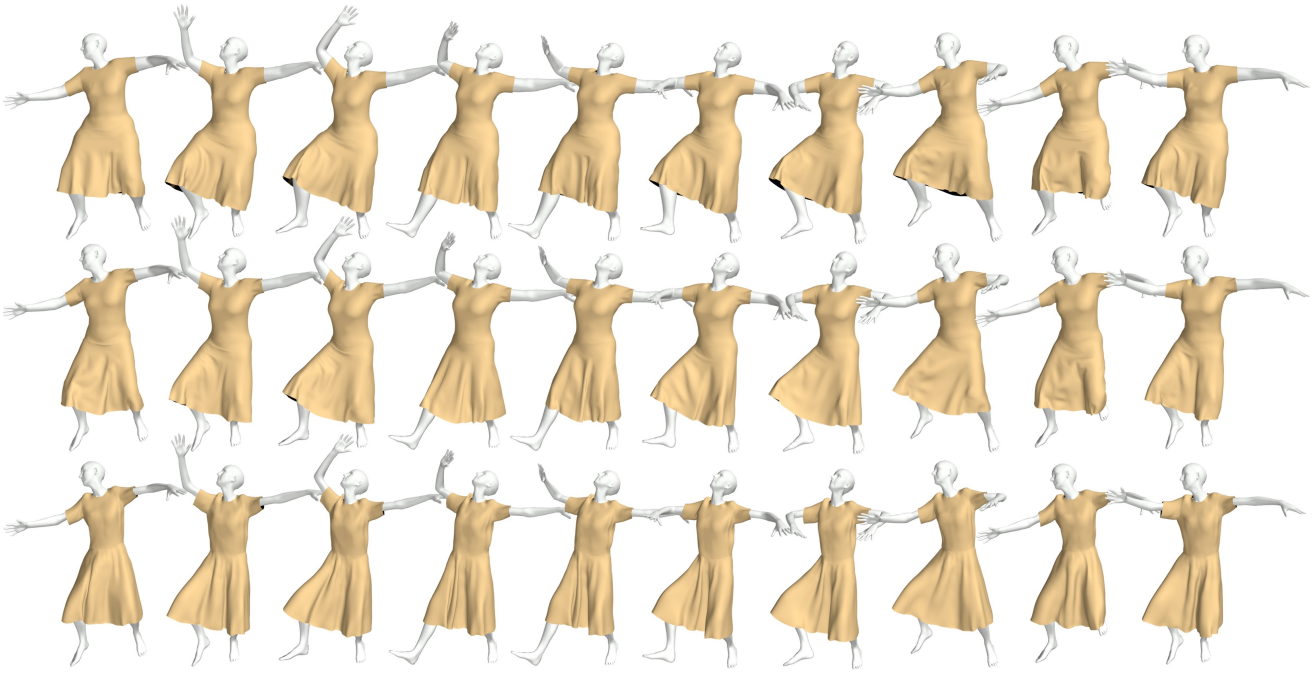


Fig. 6. Deformation results across different body shapes. Areas in close contact with thebody exhibit fewer wrinkles, while hemlines tend to display more wrinkles. The diversity anddynamics of details shows the generalization ability of our method across varying body shapes.

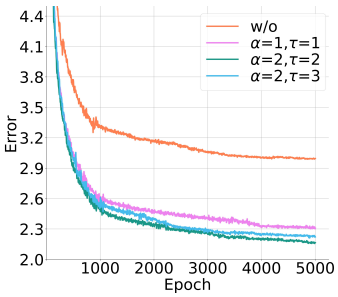


Fig. 7. Average vertex error for different network spectrum-enhancement settings. Networks without spectral enhancement significantly fall behind our method in learning deformations.

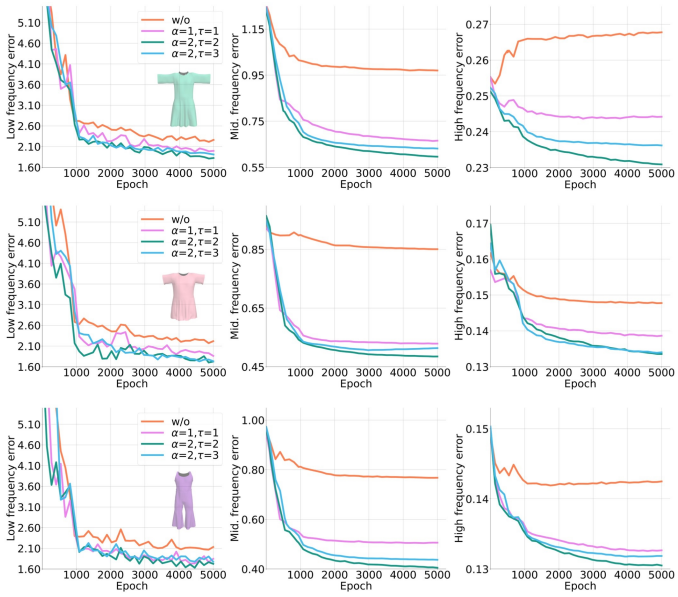


Fig. 8. Average error of three test garments in different frequency bands with various network spectrum enhancement settings. The first column illustrates the vertex distance error in the low-frequency components, comparing the prediction results with those of PBS. The second column corresponds to the middle-frequency components, while the last column corresponds to the high-frequency components.

learning process by affecting the distribution of parameters in the hidden layers. Conversely, setting these values too low causes the model to focus predominantly on a very narrow, low-frequency range of the data. This limited focus hinders the ability of the model to accurately learn complex deformation, thus reducing its effectiveness. Through our experiments, we find that appropriate parameterization improves the model precision and is crucial for achieving garment deformations that closely align with physics-based behavior.

Effect of SEGA parameters on different frequency components. Next, to emphasize the ability of our proposed method in learning the frequency component related to intricate details, we categorize the test garment mesh into three distinct frequency bands: low, medium and high, following the frequency decomposition method outlined in Fig. 2. We evaluate the model's inference performance on these bands individually. To do this, we save the network weights acquired at 50-epoch intervals during training and utilize them to predict deformations for each frequency part of the test garments. As depicted in Fig. 8, the results without spectrum enhancement strategy are worse than any of SEGA solutions. The learning of low-frequency components appears consistent using SEGA with different enhancement settings, indicating comparable efficiency and effectiveness. For middle and high-frequency components, using the original network without spectrum enhancement reveals serious

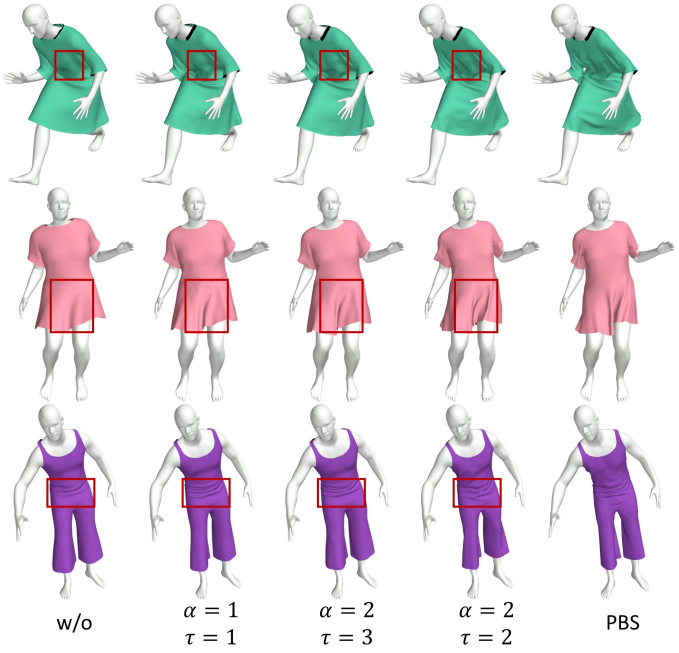


Fig. 9. Qualitative results with different network spectrum enhancement settings. The red frames in the first column highlight smoother regions in the original network without spectral enhancement. The proposed SEGA improves in generating rich details, with $\alpha = 2, \tau = 2$ setting performing the best.

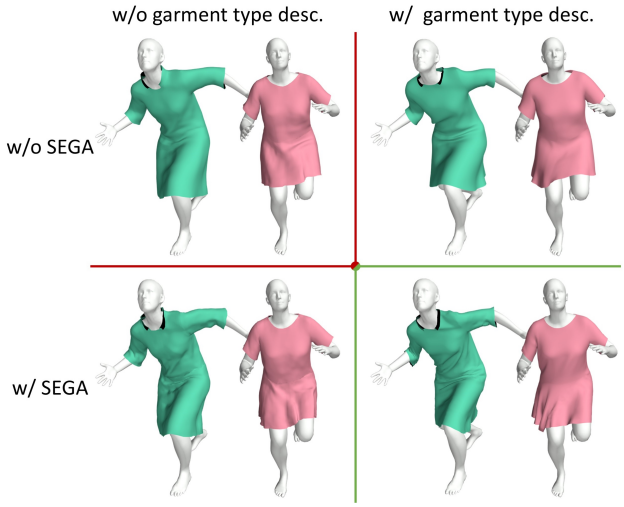


Fig. 10. Isolated validation of spectrum-related factors on deformation: the SEGA and thegarment type descriptor. We remove (w/o) and retain (w/) them separately.

spectral bias, hindering its ability to effectively capture detailed deformations. The situation becomes more pronounced, especially in the cases of dress clothing, which exhibit complex dynamic deformations. Here, errors in the middle and high-frequency components persistently remain high throughout the learning process. In contrast, configuring SEGA with appropriate parameters ($\alpha = 2, \tau = 2$) successfully achieves low errors in both the middle and high-frequency components. This facilitates comprehensive learning of both global and local deformation information.

Qualitative results are shown in Fig. 9. It can be observed that the visual fidelity of garment deformation achieved through our method closely resembles with that of PBS. Notably, our method excels in generating vivid and natural wrinkles across various regions, as highlighted by the red frame, under different motion conditions. More experimental results can be found in the supplementary materials and video.

TABLE 2
Quantitative evaluation of vertex distance error across spectrum-related factors. It presents the errors in low, middle, and high frequency components (low/middle/high) for four different cases.

	w/o garment type desc.	w/ garment type desc.
w/o SEGA	2.89 / 1.06 / 0.24	2.25 / 0.88 / 0.21
w/ SEGA	2.62 / 0.76 / 0.19	**1.81** / **0.51** / **0.18**

Independent validation of spectrum-related factors. In our framework, alongside SEGA, we include a garment type descriptor derived from the spectral decomposition of the affinity matrix. Given the spectral nature of both components, we employ conditional isolation validation to evaluate the independent contribution of each component. Specifically, we either remove or retain each factor to isolate its effects: removing SEGA indicates the utilization of the original GAT, whereas removing the garment type descriptor eliminates the extraction of garment type knowledge, relying solely on the information represented in the garment graph. Fig. 10 depicts these four scenarios. Viewing from a vertical perspective, SEGA significantly enhances the mid- and high-frequency details of the garment, notably improving the depiction of folds. However, in the absence of a garment type descriptor, the accuracy and plausibility of these details may be compromised. From a horizontal perspective, the garment type descriptor delivers global information that primarily influences the overall shape of the garment, such as the direction of the hem of the pink dress. Overall, our complete method, as illustrated in the lower-right corner of Fig. 10, integrates both global and local information, achieving a more realistic representation of garment behaviors.

Table 2 presents the average vertex distance error for four different cases. The quantitative results indicate that the two spectrum-related factors produce a mixed influence across all three frequency ranges of garment deformation. In particular, the low-frequency component of the garment exhibits a larger deviation when the garment type descriptor is absent. This is because the descriptor provides foundational information about the garment type, and its absence results in a lack of global responsiveness. In addition, the mid-frequency portion of the deformation is more significantly influenced in scenarios with and without SEGA. This experiment highlights the critical need for incorporating both spectral components to achieve precise garment deformation. Omitting either component compromises the accuracy and realism of the simulation.

5.3 Evaluation on Temporal Skinning Weights

To demonstrate the effectiveness of our proposed temporal skinning weights, we perform ablation experiments on key components for weight calculation. Table 3 presents a

TABLE 3
Quantitative ablation study on temporal skinning weights. First, we verify the temporality of the skinning weights by removing a motion processor and applying MLP to handle the current pose features. Then, we sequentially verify the target-awareness by removing the garment type descriptor, applying graph pooling instead of our garment type descriptor, and using the garment type descriptor based on Euclidean distance. The values before and after the slash symbol indicate the average vertex distance (cm), average angular deviation (°) of vertex normals, and average edge length (mm) between predictions and PBS.

Methods	Long Dress	Short Dress	Long Jumpsuit	Short Jumpsuit
w/o motion processor	3.28 / 19.7 / 0.66	3.15 / 18.8 / 0.54	3.02 / 17.6 / 0.55	3.08 / 18.1 / 0.51
w/o garment type parser	3.56 / 21.4 / 0.69	3.32 / 20.9 / 0.59	3.35 / 18.9 / 0.61	3.28 / 21.3 / 0.56
Graph pooling	3.04 / 18.2 / 0.54	2.89 / 16.1 / 0.51	2.78 / 14.6 / 0.46	2.73 / 14.2 / 0.47
Euclidean descriptor	3.16 / 18.8 / 0.59	3.08 / 16.4 / 0.52	2.88 / 15.8 / 0.48	2.79 / 15.5 / 0.44
Ours	**2.46** / **14.2** / **0.42**	**2.37** / **13.3** / **0.27**	**2.20** / **10.5** / **0.30**	**2.13** / **11.7** / **0.25**

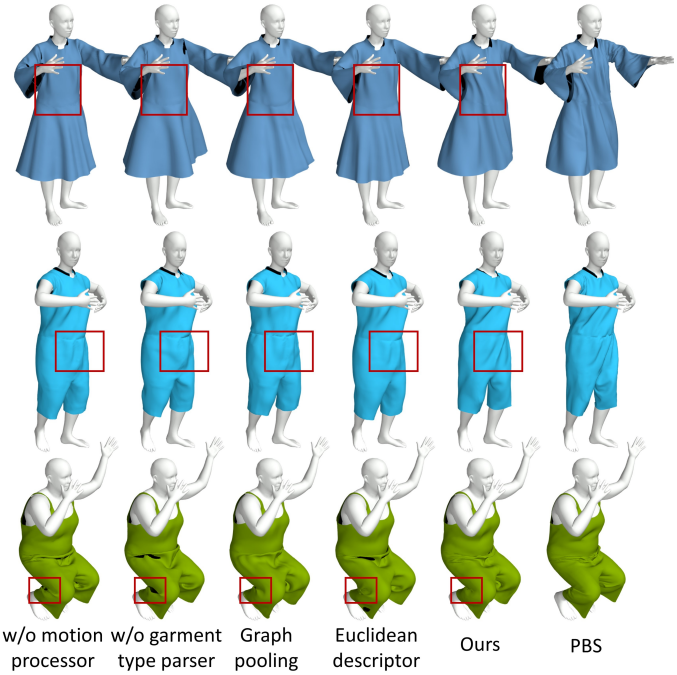


Fig. 11. Qualitative ablation study on temporal skinning weights. Note the different deformation trends and fold details in the various alternatives. The deformations generated using our full method are of the highest quality and closer to the physics-based simulation.

quantitative analysis of five different scenarios, measured in terms of average vertex distance and facet angular deviation compared to the PBS data. The first row corresponds to motion features processed by the MLP for the current frame pose, without considering the previous motion state. This analysis verifies the significance of the temporal nature of the skinning weights. Subsequently, we experiment by removing the garment type parser, causing the resulting graph features to lose the global knowledge of the garment type. This change heavily hampers the model's ability to generalize effectively. It becomes evident that both of the aforementioned alternatives yield deteriorated deformation outcomes, marked by increased prediction errors. Next, we implement a global graph pooling for global garment feature extraction, as stated in [60]. This strategy moderately enhances prediction accuracy compared to cases where no global information is available. Additionally, we maintain the network structure without changes and substitute Euclidean distance for geodesic distance to construct the affinity matrix, resulting in new garment type descriptor. However, due to the limitations of this global description in maintaining invariance and robustness in the presence of garment bending, it results in deformation errors that remain large. The final row corresponds to our full method, which attains the lowest error, providing further evidence of the beneficial nature of the proposed target-aware temporal skinning weights.

We further report the qualitative comparison in Fig. 11. As observed, when the motion processor f^{mo1} is absent, the generated skinning weights contain information solely about the current pose. This results in a weaker representation of dynamic deformation, as highlighted in the red frame in the first row. It is worth noting that we have only removed f^{mo1} responsible for generating skinning weights, while the other motion processor f^{mo2} for generating the blend shape remains unchanged. This allows the blend shapes to compensate for some deformation deviations arising from dynamics. The case where both motion processors are removed will serve as a baseline and be compared in the Sec. 5.4. Additionally, in the absence of garment type descriptor, garments appear smoother and lack fine details. Some generated wrinkles appear unusual and do not faithfully capture the clothing inherent features. To address garment meshes of diverse types, encoding garment types with a global graph pooling is a common strategy. However, this approach may introduce a quality gap when compared to physics-based simulations, because compressing the vertices in spatial domain does not accurately characterize different garments. Notably, the use of Euclidean-based descriptors leads to apparent collision artifacts in the results. This may be because the bending-invariant geodesic distance is more suitable for describing garment types and shapes than the spatial distance. In summary, our proposed target-aware temporal skinning weights excel not only in terms of accuracy but also in delivering qualitatively superior outcome compared to alternative solutions.

5.4 Comparison to State-of-the-Art Methods

We evaluate our method against recent state-of-the-art learning-based deformation approaches, mainly including SNUG [61], NCS [12], SwinGar [58], and HOOD [63]. We also implement a plain GAT-based network as a baseline for comparison.

Replication details. We provide a thorough discussion of essential techniques and implementation details employed when replicating other methods. These techniques

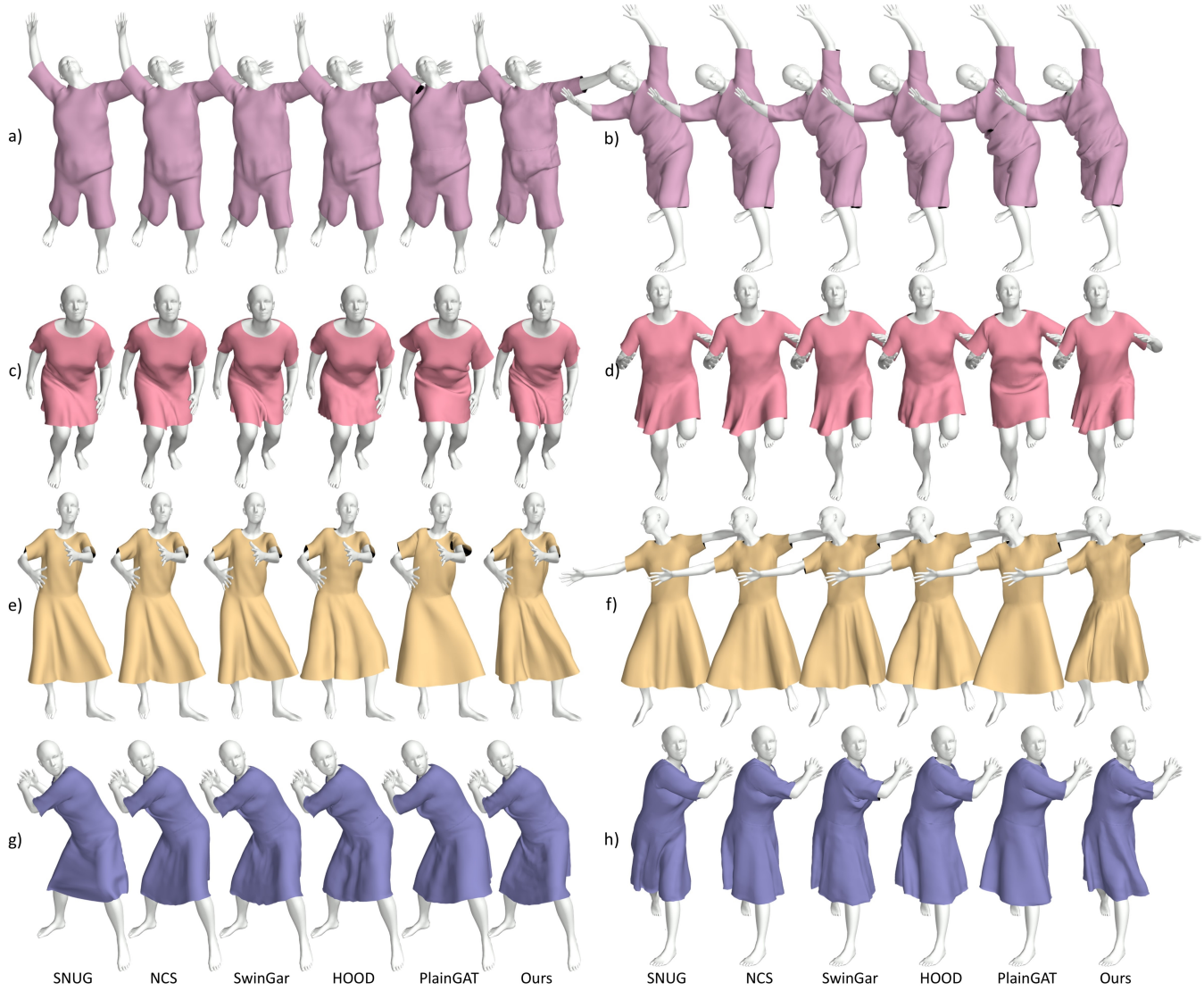


Fig. 12. Qualitative comparison for different state-of-the-art methods with different kinds of motions. Our approach is capable of generating high quality deformations with fine-scale details.

TABLE 4
Comparison with state-of-the-art methods in terms of dynamic effects, model generalization, batch support, and running time. Running performance is reported as frames per second (fps). Notably, except for HOOD [63], all other methods achieve hard real-time performance.

Methods	Dynamic	Generalization	Batch Support	Speed
SNUG [61]	✓	✗	✓	1340.4
NCS [12]	✓	✗	✓	853.9
SwinGar [58]	✓	Δ	✓	307.8
HOOD [63]	✓	✓	✗	13.1
PlainGAT	✗	✓	✓	638.5
Ours	✓	✓	✓	503.4

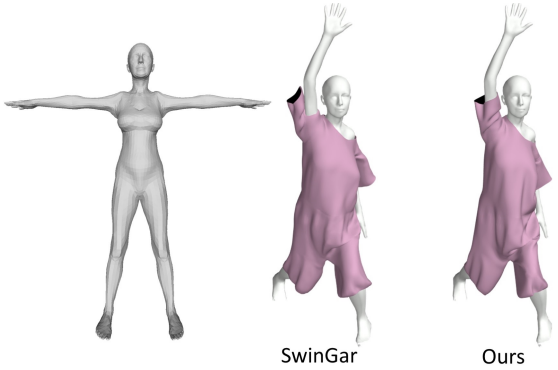


Fig. 13. Failure case of SwinGar. When the human body becomes narrower, not commonly seen in the dataset, SwinGar may exhibit a distinct interpenetration.

could be crucial for successfully reproducing the best results of previous studies.

In the case of SNUG, we find the deformation results may be ill-posed using the original inertial force loss, where $\mathcal{M}^{t-1}$ and $\mathcal{M}^{t-2}$ are calculated from $\{\theta_{t-1}, \theta_{t-2}\}$ and $\{\theta_{t-2}\}$, respectively. Based on the original idea presented in their paper [61] and related discussions in [12], we disable the gradient of $\mathcal{M}^{t-1} - \mathcal{M}^{t-2}$ when computing inertia

force which means the gradients of the loss calculation point inwards closing to the pivot, allowing SNUG to better simulate dynamics.

In NCS implementation, we find that the technique of transferring and smoothing body blend weights significantly influences training convergence and deformation effects. For longer garments such as long skirts, we take the body joints around the feet into consideration when transferring body blend weights, enhancing the blend skinning effect. Additionally, several unsupervised loss functions integrated into NCS exhibit a wide value range, requiring flexible balancing during training, depending on the garment type. In practice, especially for trousers and long skirts, empirical evidence suggests that initially minimizing the influence of collision and inertial force losses (from around $1/10$ unit) and gradually increasing their impact (up to around unit) in line with stretch and shear conditions effectively enhance the dynamic effects of complex garments and improve convergence stability.

The stage2 of SwinGar, designed to generate detailed deformations, substantially inflates GPU resources and necessitates manual memory management. Without implementing theoretical enhancements, it appears unfeasible to simplify the network structure within the SwinGar framework without compromising its detail generation ability. For HOOD, the primary challenge lies in their assignment of several features and constraints to a single vertex, which necessitates the use of heterogeneous graph structures and requires ongoing adjustments during the training process. Consequently, their method lacks support for batch inference within current deep learning frameworks, *i.e.*, only one pose sequence is processed in a batch. The cloth model is the Saint Venant Kirchhoff (StVK) elastic material model for SNUG, NCS, and HOOD. For the PlainGAT baseline, we use our network framework with the original GAT block and replace the two motion processors with MLP-based processors similar to [60], consequently lacking the assurance of fine-grained details and cloth dynamics.

Method capacity. A comprehensive analysis of deformation characteristics, model generalization to new targets, batch support, and timing performance is provided in Table 4. The check mark indicates full ability, X mark signifies a lack of ability, and triangle suggests a limited ability. Our method stands out by achieving detailed and realistic clothing deformations for arbitrary garments, irrespective of their topologies and vertex counts. This generalization enhances the applicability of our approach in scenarios demanding diverse garment types, like virtual try-on.

In our comparative analysis of different methods' running times, we measure the average time taken to transform raw pose data (*e.g.*, axis-angles or quaternions) into deformed body and garment meshes within the test dataset, parallelized on GPU. This provides an accurate representation of the expected running times in real applications. SNUG emerges as the fastest due to its concise, lightweight network design, with most computations occurring within linear operations and temporal processing. Despite using identical neural network structures, inputs, and outputs as in the original NCS implementation, we note a runtime increase of over 30%. This discrepancy may be attributed to the backend architecture of deep learning frameworks, leading us to directly report the time performance from the original paper.

Moreover, SNUG and NCS, which do not account for garment generalization, exhibit faster performance and consume less computing resources than other methods. This leads us to the case of SwinGar, which, due to its intricate detail-adding graph processing modules, may struggle to achieve real-time performance on typical computing resources with 8 or 12GB GPU memory given its substantial average GPU footprint of 10.15GB. Meanwhile, the GPU footprints of HOOD and our method, both capable of garment generalization, are significantly lower at 3.67 and 4.82GB, respectively. However, HOOD's inability to process data sequences in batch restricts it from achieving hard real-time performance. Considering that high-quality physics simulations can achieve dozens of frames per second, achieving only a dozen fps diminishes the significance of employing neural networks. In response to these constraints, by enhancing the spectral learning capability of parameterized layers, our method constructs an efficient and relatively lightweight network, achieving a performance comparable to that of NCS.

Qualitative evaluation. Qualitative results are included in Fig. 12. We show samples for different body motions: touch high (samples *a* and *b*), jumping (samples *c* and *d*), dancing (samples *e* and *f*), and swinging (samples *g* and *h*).

Among the five methods we compared, PlainGAT demonstrates the poorest performance. While it utilizes a skinning pipeline, it lacks mechanisms for detail enhancement and temporal information integration, which results in deficiencies in both dynamics and detail.

SNUG, NCS and SwinGar exhibit relatively acceptable results. However, in scenarios involving rapid movements with loose-fitting garments, the fixed blend weights of SNUG results in a more body-flattering but dynamically constrained effect around the loose areas of the dress hemline, as demonstrated in samples *g* and *h*. NCS demonstrates enhanced dynamics in deformation yet falls short in capturing detailed wrinkles, as evidenced in samples *a*, *c* and *h*. Central to this approach is an unsupervised learning framework inspired by physically based simulation. Although the correct implementation of this model ensures a meaningful capture of temporal information, it still does not break through the spectral limitations inherent in neural networks, resulting in an inability to accurately estimate realistic details. Moreover, both SNUG and NCS models are garment-specific, requiring the training of a new model for each garment type. This limitation narrows their applicability across different scenarios.

SwinGar, on the other hand, generally produces realistic results for unseen garments. However, its performance tends to degrade with extreme body shapes or poses due to its failure to adjust skinning weights for target changes. For example, substituting a thin human body with narrower shoulders in samples *a* and *b* results in sleeves deforming through the arms, as shown in Fig. 13. HOOD involves adding details on a coarse basis, where the coarse deformation is derived from body weights. However, this additive processing can result in certain garment regions lacking realism, as seen in the artificial appearance of hemline area in sample *c* and *d*. We speculate that this may be due to a

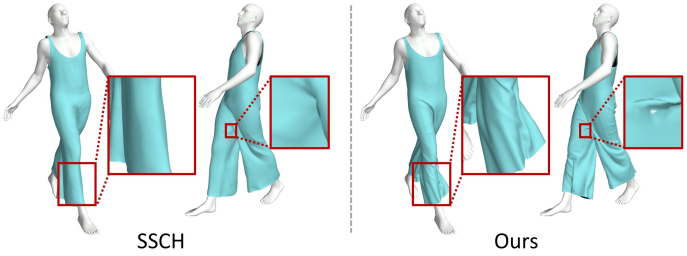


Fig. 14. Comparison to SSCH [35]. Our approach occasionally encounters collision issues; however, it produces more natural deformations with detailed wrinkles.

lack of consistency constraints on acceleration predictions.

In contrast, our method exhibits enhanced flexibility in generating dynamic wrinkles in the hem region, thanks to our two main contributions: the introduction of a spectrum-enhanced strategy and the target-aware temporal skinning weights. Additionally, we incorporate a partially unsupervised loss, which together ensure superior deformation results. More results of continuous frames can be found in the supplementary video. Upon evaluating a variety of motions and garments, our method consistently ensures hard real-time garment deformation through a unified model, while delivering high-quality details and dynamic effects.

Collision discussion. As shown in the above examples, our approach effectively generates reliable deformations in most test cases. However, even with collision constraints, it does not ensure complete collision avoidance, especially when different body parts are in close proximity. On the other hand, the SSCH method [35] more effectively prevents collisions by utilizing a diffused human model. While this approach offers better collision prevention, its reliance on a diffuse representation restricts the range of dynamic deformations that can be captured during inference, often leading to poorer dynamics and less richness in detail.

6 Conclusions and Future Work

We presented a graph learning-based framework for 3D garment animation, which offers powerful generalization capabilities, enabling the efficient generation of dynamic and intricate garment deformations. We commence with an in-depth analysis of the spectral properties of neural networks and clothing deformation, culminating in the proposal of a spectrum-enhanced deformation network. This innovation amplifies the learning of clothing details and opens up new avenues for controlling spectral bias in learning mesh-based simulation. Additionally, we leverage the efficiency of blend skinning and design the target-aware temporal skinning weights. The weights incorporate multi-source information from the garment, human body, and motion sequences, thus achieving dynamic deformations. Our approach offers extensive applicability and scalability, making it adaptable to a diverse range of costume animation scenarios.

While our method demonstrates unique advancements, several limitations remain to be addressed in future research. Firstly, similar to many state-of-the-art approaches, our model has not yet resolved the self-collision problem. Self-collision may occur in extreme cases with extensive bending. A promising way for future exploration involves formulating this problem into a loss function that automatically penalizes vertices with penetrations. Secondly, concerning garment-body collisions, our method employs a loss term during optimization, which may not always be effective with unseen data. Currently, this issue is addressed through post-processing steps [36]. Future work may involve the development of an end-to-end collision-free garment deformation prediction system. Thirdly, although Blender offers the capability to automate simulations, the process of preparing diverse types of training data and filtering out problematic data still demands considerable manual effort. One of the future directions worth exploring is how to keep the benefits of straightforward convergence found in supervised learning, while also minimizing the data preparation workload, drawing inspiration from the state-of-the-art unsupervised methods [12], [61], [63]. Finally, our method is limited to SMPL models as the human body, as it relies on well-defined parameterized human bodies to obtain necessary body shape knowledge. In the future, we aim to explore more versatile description methods for both human bodies and garments.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 62402021, 62403017, U2233211), the Beijing Natural Science Foundation (Nos. 4244088, 4232017), and the JSPS KAKENHI (No. 25K15401).

References

[1] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," *Comput. Graph. Forum*, vol. 25, no. 4, pp. 809–836, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2006.01000.x

[2] J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, and L. Boissieux, "An implicit frictional contact solver for adaptive cloth simulation," *ACM Trans. Graph.*, vol. 37, no. 4, 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201308

[3] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," in *Proc. Graph. Interface*, 1989, pp. 26–33.

[4] L. Kavan, S. Collins, J. Zára, and C. O'Sullivan, "Skinning with dual quaternions," in *Proc. Symp. Interactive 3D Graph. Games*, 2007, pp. 39–46. [Online]. Available: https://doi.org/10.1145/1230100.1230107

[5] R. Y. Wang, K. Pulli, and J. Popović, "Real-time enveloping with rotational regression," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 73–es, 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276468

[6] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation," in *Proc. Annu. Conf. Comput. Graph. Interact. Tech.*, 2000, pp. 165–172. [Online]. Available: https://doi.org/10.1145/344779.344862

[7] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, 2005. [Online]. Available: https://doi.org/10.1145/1073204.1073207

[8] R. Vidaurre, I. Santesteban, E. Garces, and D. Casas, "Fully convolutional graph neural networks for parametric virtual try-on," *Comput. Graph. Forum*, vol. 39, no. 8, pp. 145–156, 2020.

[9] J. Wu, Z. Geng, H. Zhou, and R. Fedkiw, "Skinning a parameterization of three-dimensional space for neural network cloth," *CoRR*, vol. abs/2006.04874, 2020. [Online]. Available: https://arxiv.org/abs/2006.04874

[10] T. Li, R. Shi, and T. Kanai, "Detail-aware deep clothing animations infused with multi-source attributes," *Comput. Graph. Forum*, vol. 42, no. 1, pp. 231–244, 2023.

[11] H. Bertiche, M. Madadi, and S. Escalera, "PBNS: Physically based neural simulation for unsupervised garment pose space deformation," *ACM Trans. Graph.*, vol. 40, no. 6, 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480479

[12] ——, "Neural cloth simulation," *ACM Trans. Graph.*, vol. 41, no. 6, 2022. [Online]. Available: https://doi.org/10.1145/3550454.3555491

[13] X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behaviour," in *Proc. Graph. Interface*, 1995, pp. 147–154. [Online]. Available: http://graphicsinterface.org/wp-content/uploads/gi1995-17.pdf

[14] K.-J. Choi and H.-S. Ko, "Stable but responsive cloth," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 604–611, jul 2002. [Online]. Available: https://doi.org/10.1145/566654.566624

[15] R. Narain, A. Samii, and J. F. O'Brien, "Adaptive anisotropic remeshing for cloth simulation," *ACM Trans. Graph.*, vol. 31, no. 6, 2012. [Online]. Available: https://doi.org/10.1145/2366145.2366171

[16] C. Jiang, T. Gast, and J. Teran, "Anisotropic elastoplasticity for cloth, knit and hair frictional contact," *ACM Trans. Graph.*, vol. 36, no. 4, 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073623

[17] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *J. Vis. Commun. Image Representation*, vol. 18, no. 2, pp. 109–118, 2007. [Online]. Available: https://doi.org/10.1016/j.jvcir.2007.01.005

[18] M. Müller, "Hierarchical Position Based Dynamics," in *Workshop Virtual Real. Interact. Phys. Simul.*, 2008.

[19] M. Macklin, M. Müller, and N. Chentanez, "XPBD: Position-based simulation of compliant constrained dynamics," in *Proc. Int. Conf. Motion Games*, 2016, pp. 49–54. [Online]. Available: https://doi.org/10.1145/2994258.2994272

[20] G. Cirio, J. Lopez-Moreno, D. Miraut, and M. A. Otaduy, "Yarn-level simulation of woven cloth," *ACM Trans. Graph.*, vol. 33, no. 6, 2014. [Online]. Available: https://doi.org/10.1145/2661229.2661279

[21] L. Wu, B. Wu, Y. Yang, and H. Wang, "A safe and fast repulsion method for GPU-based cloth self collisions," *ACM Trans. Graph.*, vol. 40, no. 1, 2020. [Online]. Available: https://doi.org/10.1145/3430025

[22] M. Tang, H. Wang, L. Tang, R. Tong, and D. Manocha, "CAMA: Contact-aware matrix assembly with unified collision handling for GPU-based cloth simulation," *Comput. Graph. Forum*, vol. 35, no. 2, pp. 511–521, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12851

[23] M. Müller and N. Chentanez, "Wrinkle Meshes," in *ACM SIGGRAPH Symp. Comput. Animat.*, 2010, pp. 85–91. [Online]. Available: https://doi.org/10.2312/SCA/SCA10/085-091

[24] Z. Chen, H.-Y. Chen, D. M. Kaufman, M. Skouras, and E. Vouga, "Fine wrinkling on coarsely meshed thin shells," *ACM Trans. Graph.*, vol. 40, no. 5, 2021. [Online]. Available: https://doi.org/10.1145/3462758

[25] H. Wang, "GPU-based simulation of cloth wrinkles at submillimeter levels," *ACM Trans. Graph.*, vol. 40, no. 4, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459787

[26] Z. Pan, H. Bao, and J. Huang, "Subspace dynamic simulation using rotation-strain coordinates," *ACM Trans. Graph.*, vol. 34, no. 6, 2015. [Online]. Available: https://doi.org/10.1145/2816795.2818090

[27] M. Ly, J. Jouve, L. Boissieux, and F. Bertails-Descoubes, "Projective dynamics with dry frictional contact," *ACM Trans. Graph.*, vol. 39, no. 4, 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392396

[28] S. Yang, Z. Pan, T. Amert, K. Wang, L. Yu, T. Berg, and M. C. Lin, "Physics-inspired garment recovery from a single-view image," *ACM Trans. Graph.*, vol. 37, no. 5, 2018. [Online]. Available: https://doi.org/10.1145/3026479

[29] M.-H. Jeong, D.-H. Han, and H.-S. Ko, "Garment capture from a photograph," *Comput. Animat. Virtual Worlds*, vol. 26, pp. 291–300, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cav.1653

[30] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popović, and S. M. Seitz, "Estimating cloth simulation parameters from video," in *ACM SIGGRAPH Symp. Comput. Animat.*, 2003, pp. 37–51.

[31] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, and C. Theobalt, "Video-based reconstruction of animatable human characters," in *ACM SIGGRAPH Asia*, 2010. [Online]. Available: https://doi.org/10.1145/1866158.1866161

[32] S. Yang, J. Liang, and M. C. Lin, "Learning-based cloth material recovery from video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4393–4403. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.470

[33] J. Liang, M. Lin, and V. Koltun, "Differentiable cloth simulation for inverse problems," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[34] M. Mihajlovic, Y. Zhang, M. J. Black, and S. Tang, "LEAP: Learning articulated occupancy of people," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10 456–10 466.

[35] I. Santesteban, N. Thuerey, M. A. Otaduy, and D. Casas, "Self-supervised collision handling via generative 3D garment models for virtual try-on," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11 758–11 768.

[36] I. Santesteban, M. Otaduy, N. Thuerey, and D. Casas, "ULNeF: Untangled layered neural fields for mix-and-match virtual try-on," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 12 110–12 125.

[37] X. Pan, J. Mai, X. Jiang, D. Tang, J. Li, T. Shao, K. Zhou, X. Jin, and D. Manocha, "Predicting loose-fitting garment deformations using bone-driven motion networks," in *ACM SIGGRAPH*, 2022. [Online]. Available: https://doi.org/10.1145/3528233.3530709

[38] M. Zhang, D. Ceylan, and N. J. Mitra, "Motion guided deep dynamic 3D garments," *ACM Trans. Graph.*, vol. 41, no. 6, 2022. [Online]. Available: https://doi.org/10.1145/3550454.3555485

[39] A. H. Rasheed, V. Romero, F. Bertails-Descoubes, S. Wuhrer, J. Franco, and A. Lazarus, "A visual approach to measure cloth-body and cloth-cloth friction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6683–6694, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3097547

[40] P. Guan, L. Reiss, D. A. Hirshberg, A. Weiss, and M. J. Black, "DRAPE: Dressing any person," *ACM Trans. Graph.*, vol. 31, no. 4, 2012. [Online]. Available: https://doi.org/10.1145/2185520.2185531

[41] I. Santesteban, M. A. Otaduy, and D. Casas, "Learning-based animation of clothing for virtual try-on," *Comput. Graph. Forum*, vol. 38, no. 2, pp. 355–366, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13643

[42] T. Y. Wang, D. Ceylan, J. Popović, and N. J. Mitra, "Learning a shared shape space for multimodal garment design," *ACM Trans. Graph.*, vol. 37, no. 6, 2018. [Online]. Available: https://doi.org/10.1145/3272127.3275074

[43] G. Tiwari, B. L. Bhatnagar, T. Tung, and G. Pons-Moll, "SIZER: A dataset and model for parsing 3D clothing and learning size sensitive 3d clothing," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12348, 2020, pp. 1–18. [Online]. Available: https://doi.org/10.1007/978-3-030-58580-8_1

[44] C. Patel, Z. Liao, and G. Pons-Moll, "TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7363–7373.

[45] L. Liu, Y. Zheng, D. Tang, Y. Yuan, C. Fan, and K. Zhou, "NeuroSkinning: Automatic skin binding for production characters with deep graph networks," *ACM Trans. Graph.*, vol. 38, no. 4, 2019. [Online]. Available: https://doi.org/10.1145/3306346.3322969

[46] Z. Xu, Y. Zhou, E. Kalogerakis, C. Landreth, and K. Singh, "RigNet: Neural rigging for articulated characters," *ACM Trans. Graph.*, vol. 39, no. 4, 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392379

[47] P. Li, K. Aberman, R. Hanocka, L. Liu, O. Sorkine-Hornung, and B. Chen, "Learning skeletal articulations with neural blend shapes," *ACM Trans. Graph.*, vol. 40, no. 4, 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459852

[48] T. Li, R. Shi, and T. Kanai, "MultiResGNet: Approximating non-linear deformation via multi-resolution graphs," *Comput. Graph. Forum*, vol. 40, no. 2, pp. 537–548, 2021.

[49] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia, "Learning mesh-based simulation with graph networks," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=roNqYL0_XP

[50] Z. Xu, Y. Zhou, L. Yi, and E. Kalogerakis, "Morig: Motion-aware rigging of character meshes from point clouds," in *ACM SIGGRAPH Asia*, 2022. [Online]. Available: https://doi.org/10.1145/3550469.3555390

[51] Y. Li, M. Tang, Y. bo Yang, Z. Huang, R. Tong, S. Yang, Y. Li, and D. Manocha, "N-Cloth: Predicting 3D cloth deformation with mesh-based networks," *Comput. Graph. Forum*, vol. 41, no. 2, pp. 547–558, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14493

[52] Y. Gao, Z. Kuang, G. Li, P. Luo, Y. Chen, L. Lin, and W. Zhang, "Fashion retrieval via graph reasoning networks on a similarity pyramid," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7019–7034, 2023. [Online]. Available: https://doi.org/10.1109/TPAMI.2020.3025062

[53] Y. D. Li, M. Tang, X. R. Chen, Y. Yang, R. F. Tong, B. L. An, S. C. Yang, Y. Li, and Q. L. Kou, "D-cloth: Skinning-based cloth dynamic prediction with a three-stage network," *Comput. Graph. Forum*, vol. 42, no. 7, p. e14937, 2023.

[54] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black, "Learning to dress 3D people in generative clothing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6468–6477. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.00650

[55] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, 2015. [Online]. Available: https://doi.org/10.1145/2816795.2818013

[56] R. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.16

[57] E. Gundogdu, V. Constantin, S. Parashar, A. Seifoddini, M. Dang, M. Salzmann, and P. Fua, "GarNet++: Improving fast and accurate static 3D cloth draping by curvature loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 181–195, 2022.

[58] T. Li, R. Shi, Q. Zhu, and T. Kanai, "Swingar: Spectrum-inspired neural dynamic deformation for free-swinging garments," *arXiv:2308.02827*, 2023.

[59] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 1996, p. 473–479.

[60] H. Bertiche, M. Madadi, E. Tylson, and S. Escalera, "DeePSD: Automatic deep skinning and pose space deformation for 3D garment animation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 5451–5460.

[61] I. Santesteban, M. A. Otaduy, and D. Casas, "SNUG: Self-supervised neural dynamic garments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8130–8140.

[62] L. De Luigi, R. Li, B. Guillard, M. Salzmann, and P. Fua, "DrapeNet: Garment generation and self-supervised draping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1451–1460. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00146

[63] A. Grigorev, B. Thomaszewski, M. J. Black, and O. Hilliges, "HOOD: Hierarchical graphs for generalized modelling of clothing dynamics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16 965–16 974. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01627

[64] A. Mir, T. Alldieck, and G. Pons-Moll, "Learning to transfer texture from clothing images to 3D humans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7021–7032.

[65] X. Xu and C. C. Loy, "3D human texture estimation from a single image with transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 13 849–13 858.

[66] J. Wu, Y. Jin, Z. Geng, H. Zhou, and R. Fedkiw, "Recovering geometric information with learned texture perturbations," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 4, no. 3, Sep. 2021.

[67] Y. Shen, J. Liang, and M. C. Lin, "GAN-based garment generation using sewing pattern images," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 225–247. [Online]. Available: https://doi.org/10.1007/978-3-030-58523-5_14

[68] N. Jin, Y. Zhu, Z. Geng, and R. Fedkiw, "A pixel-based framework for data-driven clothing," *Comput. Graph. Forum*, vol. 39, no. 8, pp. 135–144, 2020. [Online]. Available: https://doi.org/10.1111/cgf.14108

[69] R. Yu, X. Wang, and X. Xie, "VTNFP: An image-based virtual try-on network with body and clothing feature preservation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 10 510–10 519. [Online]. Available: https://api.semanticscholar.org/CorpusID:207980566

[70] W.-L. Hsiao, I. Katsman, C.-Y. Wu, D. Parikh, and K. Grauman, "Fashion++: Minimal edits for outfit improvement," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[71] A. Grigorev, K. Iskakov, A. Ianina, R. Bashirov, I. Zakharkin, A. Vakhitov, and V. Lempitsky, "Stylepeople: A generative model of fullbody human avatars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5151–5160.

[72] R. Basri, D. W. Jacobs, Y. Kasten, and S. Kritchman, "The convergence rate of neural networks for learned functions of different frequencies," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4763–4772. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/5ac8bb8a7d745102a978c5f8ccdb61b8-Abstract.html

[73] X. Wang and M. Zhang, "How powerful are spectral graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 162, 2022, pp. 23 341–23 362. [Online]. Available: https://proceedings.mlr.press/v162/wang22am.html

[74] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-

[75] Z. Shi, P. Mettes, S. Maji, and C. G. M. Snoek, "On measuring and controlling the spectral bias of the deep image prior," *Int. J. Comput. Vis.*, vol. 130, no. 4, pp. 885–908, 2022. [Online]. Available: https://doi.org/10.1007/s11263-021-01572-7

[76] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. C. Courville, "On the spectral bias of neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 5301–5310. [Online]. Available: http://proceedings.mlr.press/v97/rahaman19a.html

[77] R. Basri, M. Galun, A. Geifman, D. W. Jacobs, Y. Kasten, and S. Kritchman, "Frequency bias in neural networks for input of non-uniform density," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 685–694. [Online]. Available: http://proceedings.mlr.press/v119/basri20a.html

[78] Z. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," *Commun. Comput. Phys.*, vol. 28, no. 5, pp. 1746–1767, 2020. [Online]. Available: http://global-sci.org/intro/article_detail/cicp/18395.html

[79] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing lipschitz continuity," *Mach. Learn.*, vol. 110, no. 2, pp. 393–416, 2021. [Online]. Available: https://doi.org/10.1007/s10994-020-05929-w

[80] X. Qi, J. Wang, Y. Chen, Y. Shi, and L. Zhang, "Lipsformer: Introducing Lipschitz continuity to vision transformers," in *Proc. Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://openreview.net/pdf?id=cHf1DcCwcH3

[81] H. Kim, G. Papamakarios, and A. Mnih, "The Lipschitz constant of self-attention," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 5562–5571. [Online]. Available: http://proceedings.mlr.press/v139/kim21i.html

[82] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2924–2932. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/109d2dd3608f669ca17920c511c2a41e-Abstract.html

[83] M. Telgarsky, "Benefits of depth in neural networks," in *Proc. Conf. Learn. Theory*, vol. 49, 2016, pp. 1517–1539. [Online]. Available: http://proceedings.mlr.press/v49/telgarsky16.html

[84] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Proc. Conf. Learn. Theory*, vol. 49, 2016, pp. 907–940. [Online]. Available: http://proceedings.mlr.press/v49/eldan16.html

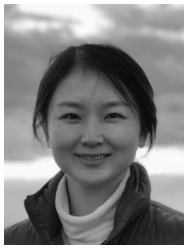
[85] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/55053683268957697aa39fba6f231c68-Abstract.html

[86] J. Zhan, L. Zhang, and J. Qiao, "Boundary consensus of networked hyperbolic systems of conservation laws," *IEEE Trans. Autom. Control*, pp. 1–16, 2025.

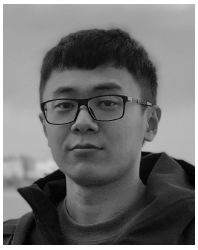
[87] R. Shi, T. Li, Y. Yamaguchi, and L. Zhang, "Traffic scene-informed attribution of autonomous driving decisions," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–12, 2025.

[88] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int.*

Conf. Learn. Representations, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ
[89] T. Li, R. Shi, and T. Kanai, "DenseGATs: A graph-attention-based network for nonlinear character deformation," in *Proc. Symp. Interactive 3D Graph. Games*, 2020, pp. 5:1–5:9.
[90] T. Ceccherini-Silberstein, F. Scarabotti, and F. Tolli, *Discrete Harmonic Analysis: Representations, Number Theory, Expanders, and the Fourier Transform*. Cambridge University Press, 2018, vol. 172.
[91] H. Federer, *Geometric measure theory*, 3rd ed. Springer Berlin, Heidelberg, 1996.
[92] Y. Katznelson, *An introduction to harmonic analysis*, 3rd ed. Cambridge University Press, 2004.
[93] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, 2004.
[94] H. Bertiche, M. Madadi, and S. Escalera, "CLOTH3D: Clothed 3D humans," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 344–359.
[95] Carnegie-Mellon, "CMU graphics lab motion capture database," http://mocap.cs.cmu.edu/, 2010, accessed: 2023.



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



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

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

Liguo Zhang received his Ph.D. degree in control theory and applications from the Beijing University of Technology (BJUT), Beijing, China, in 2006. Since 2014, he has been a Full Professor with the School of Electronic Information and Control Engineering, BJUT. He is currently the Deputy Director of the School of Information Science and Technology, BJUT. His research interests include hybrid systems, intelligent systems, and control of distributed parameter systems. He is an Associate Editor for the IMA Journal Mathematical Control and Information and the Guest Editor of the International Journal of Distributed Sensor Networks.

Takashi Kanai received his Ph.D. degree in engineering from the University of Tokyo in 1998. He is a professor in the Graduate School of Arts and Sciences, the University of Tokyo, Tokyo, Japan. His research interests include geometry processing and physics-based animation in computer graphics. He is a member of ACM, ACM SIGGRAPH, IIEEJ (the Institute of Image Electronics Engineers of Japan), and IPSJ (Information Processing Society of Japan).