PhysDrape: Learning Explicit Forces and Collision Constraints for Physically Realistic Garment Draping

Minghai Chen[1,2⋆], Mingyuan Liu[1⋆], Ning Ma[1], Jianqing Li[2], and Yuxiang Huan[1⋆⋆]

[1] Guangdong Institute of Intelligence Science and Technology, Zhuhai, Guangdong, China
[2] Macau University of Science and Technology, Macau SAR, China
yxhuan@gdiist.cn

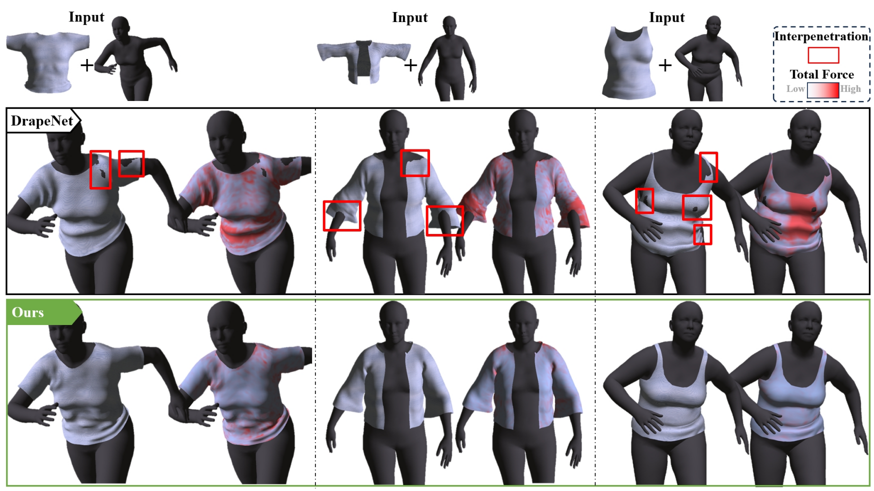

Fig. 1: Visualization results show that PhysDrape achieves more physically realistic 3D garment draping by unifying deep learning generalization with physical solver fidelity. Across different human poses and garment types, PhysDrape demonstrates quantitatively more plausible force distributions (highlighted in the heatmap) and qualitatively more realistic deformations, with fewer interpenetrations (highlighted in red boxes).

Abstract. Garment draping aims to fit clothing onto 3D human models. Existing physics-based methods are computationally expensive and struggle to integrate with general differentiable systems. In contrast, deep learning-based methods often require extensive annotations and lack explicit physical constraints, limiting their ability to model accurate details.

⋆ Equal contribution
⋆⋆ Corresponding author

We innovate by bridging physical models with deep learning and propose PhysDrape. It integrates three modules using forces as an intermediary: the force-driven GNN predicts forces at each node, the stretching solver models physical deformation, and the collision handler penalizes interpenetration. Each module includes learnable parameters to fit underlying force propagation processes and physical properties, enabling generalization to unseen templates and control over properties like stiffness. PhysDrape enables end-to-end optimization, leveraging intrinsic garment energy for physically plausible deformation via self-supervised learning. Following the previous training and evaluation protocol on CLOTH3D dataset, PhysDrape achieves lower energy scores, negligible interpenetration, more realistic visualizations, and comparable time consumption.

Keywords: Garment Draping · Neural Fabric Simulation · Physical Solver · Graph Neural Networks

1 Introduction

Garment draping involves fitting clothing onto a 3D human model in various poses to achieve realistic body conformity. This process simulates fabric deformations such as folding, stretching, and motion, which is essential for applications like virtual try-on, animation, gaming, and the metaverse [15, 40, 48].

Existing approaches primarily rely on physics-based simulations or deep learning-based regression, each of which has its limitations. Specifically, **physics-based simulations** employ multi-round solvers to model fabric properties, incurring high computational costs [1, 10, 21, 29–31, 35, 36, 42, 46, 49]. Additionally, their reliance on discrete computation hinders joint optimization with other algorithms, such as recovering 3D shapes and physical parameters from visual input, limiting their generalization across diverse scenarios. **Deep learning-based regression** methods learn from large annotated datasets and directly predict garment geometry from body poses [4, 5, 13, 14, 33, 37]. However, these methods lack explicit physical constraints, leading to unrealistic garment behavior, e.g., unnatural draping, interpenetration, and inaccurate garment details.

Although pioneering efforts integrate deep learning with physics-informed designs, they fall short of explicitly incorporating physical processes into deep learning models, and thus lack explicit mechanisms to maintain the physical fidelity of the interaction between the garment and the body. In particular, earlier works [3,38] primarily focus on incorporating physics-informed loss functions specifically tailored to individual garments. Subsequent works [9, 12, 18, 20, 45] aim to generalize the model across diverse garment topologies. Additionally, supervised postprocessors [43, 44] can be applied after coarse draping to rectify physically inconsistent regions. *However*, these methods face two main issues: 1) modeling physical constraints as loss functions lacks strict physical enforcement, leading to inaccurate results such as severe interpenetration. 2) Postprocessing fails to jointly optimize with garment draping, leading to suboptimal fabric deformation.

In this work, we propose PhysDrape, which explicitly integrates deep learning models with physical deformation models. It consists of three components: force-driven GNN, stretching solver, and collision handler. Specifically, the **force-driven GNN** estimates the forces acting on each node of the mesh, rather than the typical position regression, enabling enhanced mesh-based physical deformation that adapts to complex body geometries. The **stretching solver** iteratively deforms the garment by physically propagating forces through the mesh, with learnable parameters fitting the force propagation patterns in the fabric. The **collision handler** pushes the fabric out of the body to resolve interpenetration, while using learnable parameters to fit hard-to-model physical quantities, such as elasticity, at the body-garment interface. Notably, these three modules are learnable and enable end-to-end optimization in a self-supervised manner, using an energy-based loss function. This joint optimization of the neural network and explicit learnable physical solver not only enables generalization to unseen garments, but also allows control over physical parameters, such as stiffness, facilitating more physically plausible draping. Following previous protocols [9], PhysDrape is validated using garments and template meshes from the CLOTH3D dataset, without ground truth [2]. The results demonstrate that PhysDrape achieves better energy scores, fewer interpenetrations, more realistic visualizations, and comparable time consumption.

To summarize, our main contributions are threefold:

- We propose PhysDrape, a novel model that unifies neural networks and physical simulations through forces as an intermediary, enabling self-supervised end-to-end optimization for more physically realistic 3D garment draping.
- We propose a novel differentiable stretching and collision solver, integrated with learnable parameters to fit hard-to-measure physical quantities for physical fidelity. It enables joint optimization with neural networks, providing not only generalization to unseen garments but also explicit control over physical stiffness to simulate different fabrics.
- PhysDrape achieves state-of-the-art performance, reflected in lower energy scores, less interpenetration, more realistic visualizations, and comparable time consumption. Ablation experiments further demonstrate the effectiveness of each design.

2 Related Works

2.1 3D Garment Draping

Approaches for garment draping generally fall into two categories: physics-based simulation and deep learning-based regression. Physics-based simulation models cloth dynamics using Mass–Spring Systems (MSS) [36] or Finite Element Methods (FEM) [11], typically employing optimization-based solvers like Position-Based Dynamics (PBD) [26], Extended PBD (XPBD) [23], and Projective Dynamics (PD) [6] for stability. *However*, these methods typically rely on non-differentiable discrete projections, limiting joint optimization with other approaches and thus hindering their application in modern AI systems. While differentiable variants like DiffXPBD [41] and IPC [17] exist, they require large linear systems at each step, imposing heavy computational burdens. As a result, differentiable deep learning systems are increasingly emphasized.

Deep learning-based 3D garment draping focuses on regressing garment geometry from body poses, later incorporating physics-informed designs to improve realism. Early works generate large datasets of body-garment pairs using annotations and offline simulations to train neural networks in a supervised manner [4, 5, 13, 14, 33, 37]. These approaches are labor-intensive and suffer from a significant sim-to-real gap. Later works introduce energy-based loss functions to train neural networks, guiding the garment draping results to adhere to physical laws from an optimization perspective. For instance, SNUG [38] and PBNS [3] pioneer physics-based self-supervision by directly minimizing physical energies. Santesteban et al. [39] propose a generative framework that utilizes a diffused body representation to explicitly handle collisions. GAPS [7] introduces a collision-aware geometrical loss, allowing adaptive stretching to resolve interpenetrations. To generalize to more complex and diverse garment templates, later works optimize for various garment types and layers. DIG [19] introduces implicit Signed Distance Functions (SDF) [32] for draping, handling arbitrary geometries and reconstructing meshes with the marching cubes algorithm [16]. DrapeNet [9] integrates physics-based objectives into a generative framework based on unsigned distance functions [8, 47, 51], enabling realistic draping of diverse fabrics. ISP [18] further introduces implicit sewing patterns for complex multi-layered garments. *However*, these methods are limited by soft penalties at the loss function level and often rely on postprocessing to handle collisions instead of joint optimization, leading to a lack of reliable physical constraints and suboptimal model optimization. In contrast, our PhysDrape constructs explicit physical processes at the model level and seamlessly integrates the neural network, stretching solver, and collision handler into an end-to-end optimizable framework, further ensuring the realism of 3D garment draping. Additionally, the physical-neural integration allows the model to generalize to unseen garments and support simulation with manually specified material stiffness.

2.2 Garment-Body Interpenetration

To prevent the garment from penetrating the body geometry, existing methods primarily rely on geometric correction and penalty-based learning approaches. Geometric correction approaches [37] directly project penetrating vertices onto the body surface. This process is non-differentiable, ignores physical constraints, and hence leads to non-physical distortions. Penalty-based learning is commonly used in neural network-based methods and enables end-to-end training. For example, [3, 9, 38] employ soft collision penalties, with SENC [20] introducing a global intersection volume loss for self-collisions. ReFU [43] adds a learnable unit to predict displacement scales for SDF-based repulsion, and SAF [44] extends this correction to the triangle mesh using local features. *However*, the aforementioned methods rely on collision postprocessors, novel loss optimizations, and annotated labels for collision avoidance learning, which often result in laborious

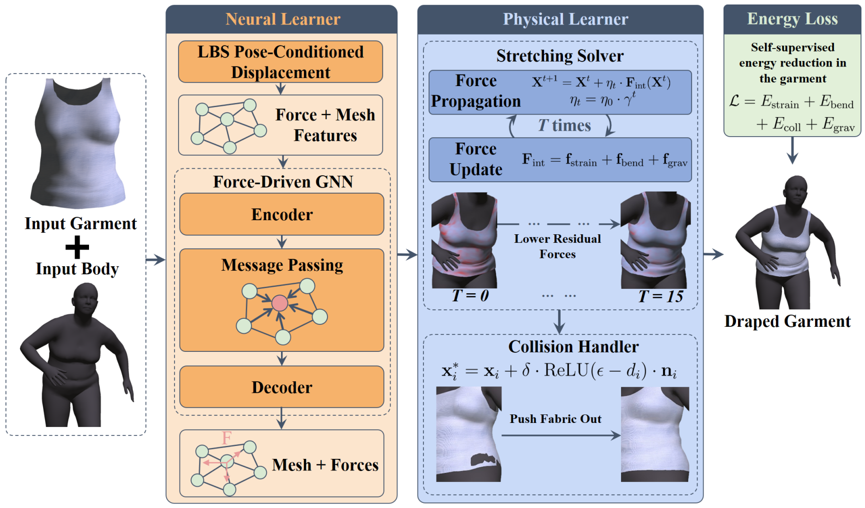

Fig. 2: PhysDrape innovatively unifies neural networks and explicit physical simulations through forces as an intermediary, enabling self-supervised end-to-end optimization for more physically realistic 3D garment draping. It integrates a force-driven GNN, a stretching solver, and a collision handler, with learnable parameters to compensate for the absence of explicit physical properties, such as fabric stiffness and contact response, thereby promoting physically plausible results.

annotations and unresolved garment force imbalances. In contrast, PhysDrape self-supervisedly learns and explicitly integrates a differentiable design into the learning system, avoiding extensive annotations while jointly optimizing penetration and fabric deformation to ensure physical fidelity.

3 Methods

3.1 Overview

We propose PhysDrape, which integrates explicit physical processes into deep learning models to achieve physically realistic 3D draping, as illustrated in Fig. 2. PhysDrape, centered around forces acting on garment mesh nodes (Sec. 3.2), consists of three main components: the force-driven GNN, the stretching solver, and the collision handler. **The force-driven GNN** (Sec. 3.3) produces a garment and derives the physical forces on the garment mesh by considering both human pose and garment shape, ensuring superior physical fidelity. **The stretching solver** (Sec. 3.4) deforms the garment based on the derived forces from the GNN, with learnable factors fitting the properties of force propagation through the fabric. **The collision handler** (Sec. 3.5) resolves overlaps and introduces learnable contact parameters, representing the physical quantities of body-garment contact. All three modules are differentiable, enabling end-to-end optimization of the entire system using self-supervision (Sec. 3.6), resulting in more realistic deformation. We adopt the SMPL [22] model to parameterize human body mesh. To formalize the internal fabric mechanics, we adopt the Saint Venant-Kirchhoff (StVK) material model [27], which is widely proven effective in virtual garment simulation and neural garment modeling [9, 12, 38].

3.2 Draping Force Formulation

To achieve physically plausible draping, we formulate the draping process as the pursuit of a quasi-static state by explicitly resolving intrinsic force in the garment. Specifically, the draped state is governed by three distinct components: 1) the strain force $\mathbf{f}_{\text{strain}}$, which penalizes in-plane stretching and shearing to preserve the static structure of the garment 2) the bending force $\mathbf{f}_{\text{bend}}$, which resists out-of-plane sharp creases to maintain surface smoothness; and 3) the gravitational force $\mathbf{f}_{\text{grav}}$, which naturally drives the drape downwards. Under the StVK material formulation, we model the in-plane strain force acting on vertex i as:

$$\mathbf{f}_{\text{strain},i} = -\sum_{T \in \mathcal{N}(i)} V_T \mathbf{P}_T \mathbf{b}_{i,T} \tag{1}$$

, where V_T denotes the reference volume (i.e., rest area multiplied by thickness) of the mesh face T. $\mathbf{P}_T$ is the first Piola-Kirchhoff stress tensor, derived as $\mathbf{P}_T = \mathbf{F}_T \mathbf{S}_T$, where $\mathbf{F}_T$ is the deformation gradient mapping the rest pose to the current state, and $\mathbf{S}_T = 2\mu\mathbf{G} + \lambda \text{Tr}(\mathbf{G})\mathbf{I}$ is the second Piola-Kirchhoff stress based on the Green-Lagrange strain $\mathbf{G}$ [27]. The Lamé constants μ and λ govern the fabric's shear and stretch stiffness, respectively. Finally, $\mathbf{b}_{i,T}$ represents the shape function gradient evaluated at the rest pose, serving as a geometric mapping that distributes the face-wise stress back to the specific vertex i as a directional nodal force.

To penalize sharp out-of-plane creases, the bending force on vertex i is defined as:

$$\mathbf{f}_{\text{bend},i} = -\sum_{e \in \mathcal{H}(i)} k_{\text{bend}} s_e \theta_e \nabla_{\mathbf{x}_i} \theta_e \tag{2}$$

, where $\mathcal{H}(i)$ denotes the set of all local hinges e (formed by adjacent triangle pairs) that involve vertex i; k_{bend} is the bending stiffness coefficient; and $s_e = l_e^2/(4A_e)$ acts as the geometric scale factor of the hinge. The term $\nabla_{\mathbf{x}_i}\theta_e$ represents the gradient of the dihedral angle θ_e with respect to the vertex i, inherently directing the restoring force along the face normals to penalize bending deformation.

Finally, to naturally drive the fabric downwards, the gravitational force is given by:

$$\mathbf{f}_{\text{grav},i} = m_i \mathbf{g} \tag{3}$$

, where m_i denotes the mass assigned to vertex i, and $\mathbf{g}$ represents the gravitational acceleration vector.

The internal active force $\mathbf{f}_{\text{int},i}$ acting on vertex i is the aggregation of all driving forces:

$$\mathbf{f}_{\text{int},i} = \mathbf{f}_{\text{strain},i} + \mathbf{f}_{\text{bend},i} + \mathbf{f}_{\text{grav},i} \quad (4)$$

PhysDrape explicitly learns to propagate and resolve these forces acting on the garment to achieve force equilibrium and produce physically plausible draping.

3.3 Force-Driven Graph Neural Network

To model the physical interaction between the human body and the fabric, we introduce the Force-Driven Graph Neural Network (GNN) to derive the garment mesh and the physical forces acting upon the garment. To initialize this process, we follow established practices [19,39] to generate a pose- and shape-conditioned deformation over the template garment $\mathbf{X}_{\text{temp}} \in \mathbb{R}^{N\times 3}$, roughly fitting it to the target human body:

$$\mathbf{X}_{\text{init}} = \mathcal{W}(\mathbf{X}_{\text{shape}}, \boldsymbol{\beta}, \boldsymbol{\theta}, \mathcal{W}_G), where\ \mathbf{X}_{\text{shape}} = \mathbf{X}_{\text{temp}} + \Delta\mathbf{X}_{\text{shape}}(\boldsymbol{\beta}) \quad (5)$$

, where $\Delta\mathbf{X}_{\text{shape}}(\boldsymbol{\beta}) \in \mathbb{R}^{N\times 3}$ represents the shape-dependent vertex displacements driven by the human body shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$. This explicitly adapts the garment template to fit the target body shape before physical refinement. It is formulated as a continuous learned function for end-to-end differentiability. The function $\mathcal{W}(\cdot)$ denotes the standard Linear Blend Skinning (LBS) algorithm [22] parameterized by the target pose $\boldsymbol{\theta} \in \mathbb{R}^{72}$. Following [19,39], both the shape displacements $\Delta\mathbf{X}_{\text{shape}}$ and the skinning weights $\mathcal{W}_G$ are formulated as continuous learned functions of the target body parameters to enable differentiable kinematic fitting. While this kinematic initialization $\mathbf{X}_{\text{init}}$ efficiently aligns the garment with the body pose, it completely lacks physical awareness, often resulting in unnatural artifacts and body-garment interpenetration. Therefore, we formulate the garment mesh as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ based on its underlying topology. Using $\mathbf{X}_{\text{init}}$ as the geometric foundation, we explicitly compute the node features $\mathbf{h}_i$ for each vertex $i \in \mathcal{V}$ and the edge features $\mathbf{e}_{ij}$ for each connected pair $(i, j) \in \mathcal{E}$ as follows:

$$\begin{aligned} \mathbf{h}_i &= [\mathbf{n}_i, \mathbf{d}_i, \mathbf{f}_{\text{int},i}, \mathbf{m}_i] \in \mathbb{R}^{11}, \\ \mathbf{e}_{ij} &= [\mathbf{x}_{ij}, \|\mathbf{x}_{ij}\|, \mathbf{x}_{ij}^0, l_{ij}^0] \in \mathbb{R}^8 \end{aligned} \quad (6)$$

Node Features. The node feature $\mathbf{h}_i$ equips the GNN with both geometric context and physical states. Specifically, $\mathbf{n}_i$ is the vertex normal for capturing local surface orientation and curvature, and $\mathbf{d}_i$ denotes the signed distance to the human body surface, enabling the network to perceive potential collisions. Crucially, we explicitly compute the internal active force $\mathbf{f}_{\text{int},i}$ (as defined in Eq. 4) at the current initialized state $\mathbf{X}_{\text{init}}$. Feeding these forces into the GNN provides the network with a direct physical gradient toward equilibrium. Finally, $\mathbf{m}_i = [\mu, \lambda, k_{\text{bend}}, m_i]$ encodes the vertex-specific material properties, comprising the Lamé coefficients for membrane stiffness, the bending coefficient, and the nodal mass.

Edge Features. To capture local topological deformations, the edge feature $\mathbf{e}_{ij}$ concatenates geometric properties in both the current deformed space ($\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and its norm $\|\mathbf{x}_{ij}\|$) and the canonical rest space ($\mathbf{x}^0_{ij}$ and the rest length l^0_{ij}). This dual-space representation explicitly allows the message passing layers to measure local stretch and orientation variations relative to the rest state.

Architecture. The GNN in PhysDrape is an encode-process-decode architecture [34]. First, the Encoder module projects the raw node features $\mathbf{h}_i$ and edge features $\mathbf{e}_{ij}$ (defined in Eq. 6) into high-dimensional latent embeddings $\mathbf{h}_i^{(0)}$ and $\mathbf{e}_{ij}^{(0)}$, respectively. This is achieved via two independent multi-layer perceptrons, denoted as $\mathrm{MLP}^{\mathrm{enc}}_{\mathrm{node}}(\cdot)$ and $\mathrm{MLP}^{\mathrm{enc}}_{\mathrm{edge}}(\cdot)$, respectively:

$$\begin{aligned} \mathbf{h}_i^{(0)} &= \mathrm{MLP}^{\mathrm{enc}}_{\mathrm{node}}(\mathbf{h}_i), \\ \mathbf{e}_{ij}^{(0)} &= \mathrm{MLP}^{\mathrm{enc}}_{\mathrm{edge}}(\mathbf{e}_{ij}) \end{aligned} \tag{7}$$

Then, the processor iteratively updates these latent representations via L message passing layers to propagate physical interactions across the garment topology. Specifically, at each layer l, an edge network $\mathrm{MLP}_{\mathrm{edge}}(\cdot)$ updates the interaction features between connected vertices, and a node network $\mathrm{MLP}_{\mathrm{node}}(\cdot)$ subsequently updates the individual vertex states by aggregating messages from their local neighborhoods $\mathcal{N}(i)$:

$$\begin{aligned} \mathbf{e}_{ij}^{(l+1)} &= \mathbf{e}_{ij}^{(l)} + \mathrm{MLP}_{\mathrm{edge}}\left(\mathbf{e}_{ij}^{(l)}, \mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right), \\ \mathbf{h}_i^{(l+1)} &= \mathbf{h}_i^{(l)} + \mathrm{MLP}_{\mathrm{node}}\left(\mathbf{h}_i^{(l)}, \sum\nolimits_{j \in \mathcal{N}(i)} \mathbf{e}_{ij}^{(l+1)}\right) \end{aligned} \tag{8}$$

Finally, a decoder $\mathrm{MLP}^{\mathrm{dec}}(\cdot)$ projects the evolved node embeddings $\mathbf{h}_i^{(L)}$ back into the physical space to produce the final network predictions:

$$\hat{\mathbf{X}}_{\mathrm{pred},i}, \hat{\mathbf{f}}_i = \mathrm{MLP}^{\mathrm{dec}}(\mathbf{h}_i^{(L)}) \tag{9}$$

Notably, rather than merely predicting geometric displacements, our GNN simultaneously infers the garment mesh and the forces acting upon it.

3.4 Learnable Stretching Solver

We employ a learnable stretching solver $\mathcal{S}_{\mathrm{force}}$ to propagate the forces acting upon the garment mesh and to relax the garment into an equilibrium state. Starting from the output of the force-driven GNN, the solver performs T explicit forward simulation steps to propagate forces and update the garment mesh. Specifically, at each iteration $t \in [0, T-1]$, we calculate the updated internal active forces by evaluating $\mathbf{F}_{\mathrm{int}}(\cdot)$ at the current mesh geometry $\mathbf{X}^{(t)}$, following the mechanical formulations defined in Sec. 3.2. Subsequently, the vertex positions are updated along the direction of these forces to propagate the physical stretch:

$$\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} + \hat{\eta} \cdot \mathbf{F}_{\text{int}}(\mathbf{X}^{(t)}), \quad \hat{\eta} = \eta \cdot \gamma^{(t)} \tag{10}$$

, where η is the learnable compliance parameter (inverse stiffness), and $\gamma \in (0, 1)$ is a decay rate to stabilize the force solving process as the system approaches equilibrium.

3.5 Collision Handler

To enforce the non-penetration constraint, we pass $\mathbf{X}^{(T)}$ from the stretching solver to a differentiable collision handler $\mathcal{P}_{\text{proj}}$. For each candidate vertex $\mathbf{x}_i \in \mathbf{X}^{(T)}$, we identify its nearest neighbor $\mathbf{p}_i$ on the body surface and the corresponding outward surface normal $\mathbf{n}_i$. The signed distance from the garment vertex to the body is computed as $d_i = (\mathbf{x}_i - \mathbf{p}_i) \cdot \mathbf{n}_i$.

A collision occurs if this distance falls below a predefined threshold ($d_i < \epsilon$). We actively resolve such penetrations by projecting the vertex outward along the body normal $\mathbf{n}_i$:

$$\mathbf{x}_i^* = \mathbf{x}_i + \delta \cdot \text{ReLU}(\epsilon - d_i) \cdot \mathbf{n}_i \tag{11}$$

, where $\mathbf{x}_i^*$ represents the final collision-free vertex position, and ϵ is the margin. To enforce the non-penetration constraint, we introduce a learnable projection scalar $\delta \geq 1$. By adaptively pushing vertices further outward, δ creates a geometric buffer that ensures the attached continuous fabric faces are lifted completely clear of the body. Unlike taking collision avoidance as post-processing [37], our projection is embedded within the end-to-end learning pipeline. This integration ensures that the force-driven GNN and the stretching solver receive direct gradient feedback from the collision constraints, driving the entire network to minimize internal forces and seek physical equilibrium strictly within a valid, interpenetration-free state space.

3.6 Training Objective

PhysDrape is trained in a self-supervised manner, eliminating the need for ground truth 3D draped garments, which enhances its generalization capability. We use the energy-based loss function proposed in prior work [9] as the optimization objective. Specifically, the overall formulation of the physics-informed training objective is as follows:

$$\mathcal{L} = E_{\text{strain}} + E_{\text{bend}} + E_{\text{coll}} + E_{\text{grav}} \tag{12}$$

The total loss consists of four energy terms defined on the garment triangle mesh: in-plane fabric stretching (E_{strain}), out-of-plane sharp folding (E_{bend}), body-garment collisions (E_{coll}), and gravity (E_{grav}). Together, these terms synthesize a physically plausible draping state without requiring annotated 3D supervision. Building upon this loss function, this work focuses on integrating physical solvers with neural networks to model explicit and learnable physical processes beyond conventional loss optimization.

4 Experiments and Results

4.1 Datasets and Implementation Details

Datasets. CLOTH3D dataset [2] is used for training and evaluation, following the experimental protocol in DrapeNet [9], with a focus on upper-body garments, including t-shirts, shirts, tank tops, etc. For training, we randomly select 600 top garments and use template meshes in their canonical state (T-pose on an average body shape). Notably, this work follows a self-supervised design, requiring no additional annotations for training. Therefore, we do not use the simulated deformations of the selected garments from the dataset as supervisory signals. For testing, we use a separate set of 30 unseen top garments to better assess the generalization ability of the model to different garments. Body poses are randomly sampled from the AMASS dataset [24], and the shape parameter $\boldsymbol{\beta}$ is uniformly sampled from $[-3, 3]^{10}$, consistent with [9, 38].

Implementation Details. In PhysDrape, the physics-informed GNN follows an encode-process-decode architecture [34]. The encoder maps the input features to a 128-dimensional latent space. The processor comprises 16 message-passing blocks, with each block employing a 2-layer MLP with ReLU activation and Layer Normalization. The graph connectivity is defined by the mesh topology of the garment, augmented with bi-directional edges.

PhysDrape is trained on a single NVIDIA V100 GPU for 150,000 iterations using the Adam optimizer [50] with an initial learning rate of 5×10^{-5}. To ensure stable force propagation and physical fidelity, the learnable parameters are carefully constrained. Specifically, the per-vertex material parameters (μ, λ, k_{bend}, m_i) are uniformly sampled from predefined physical ranges, while the global solver parameters (η, γ) are heuristically initialized. During training, all physical parameters are strictly clipped to these valid ranges and regularized via an L_2 penalty to prevent extreme values. As shown in our ablations, this constrained optimization is crucial for steady energy reduction and robust collision handling. Consistent with the self-supervised protocol of DrapeNet [9], training is conducted exclusively on static garment meshes in the canonical T-pose. Although our method supports variable material properties (e.g., stiffness and density), these parameters are aligned with the fixed settings from DrapeNet during quantitative comparison to ensure a fair evaluation. End-to-end training is performed with the learnable solver set to $T = 3$ iterations, while for evaluation, we increase this to $T = 15$ to demonstrate the consistent performance improvements achieved with additional solver steps.

Evaluation Metrics. To quantitatively evaluate garment draping, we use common physics-based energy metrics [9], including strain (E_{strain}) for surface stretching and bending (E_{bend}) for folding smoothness. Lower energy values indicate a more relaxed and physically plausible state. Additionally, we assess collision handling using the Body-to-Garment (B2G) ratio [9, 19], which measures interpenetration by calculating the ratio of garment surface area penetrating the body

Table 1: PhysDrape quantitatively outperforms previous state-of-the-art methods, as demonstrated by lower physical energy metrics (E_{strain}, E_{bend}) and interpenetration ratios (B2G) on CLOTH3D testing set.

Method	$E_{\text{strain}} \downarrow$	$E_{\text{bend}} \downarrow$	B2G % $\downarrow$
DeePSD [4]	7.22	0.01	7.2
DIG [19]	6.32	0.01	1.8
DrapeNet [9]	0.43	0.01	0.9
PhysDrape ($T = 3$)	0.20	**0.004**	**0.05**
PhysDrape ($T = 15$)	**0.15**	**0.004**	**0.05**

to total garment area. These metrics directly measure the physical plausibility and geometric validity of the draped garments.

4.2 Comparisons with State-of-the-Arts

Quantitative Comparison. PhysDrape outperforms three state-of-the-art approaches: DeePSD [4], DIG [19], and DrapeNet [9] on the same testing set from CLOTH3D, as reported in Table 1. As shown in the results, baseline methods DeePSD and DIG exhibit high strain energies (> 6.0) and noticeable B2G ratios (7.2% and 1.8%, respectively), while DrapeNet retains a B2G ratio of 0.9% mainly through its self-supervised collision avoidance design. The proposed PhysDrape ($T = 15$) significantly outperforms these baselines, as demonstrated by lower fabric energies (0.15 strain energy, 0.004 bending energy) and significantly reduced interpenetration at a negligible level (B2G $\approx 0.05\%$). The superiority of PhysDrape is mainly attributed to the integration of an explicit physical solver within the deep learning model, ensuring physical plausibility.

We note that other garment draping methods cannot be fairly compared due to being either template-specific or dynamic. Template-specific methods, such as SNUG [38] and GAPS [7], require learning specific garment templates, whereas PhysDrape does not rely on template learning and can generalize to unseen templates, making a fair comparison infeasible. Dynamic methods, such as Hood [12] and SENC [20], simulate fabric deformation as the body moves from a T-pose to a given pose, with motion patterns affecting the draping outcome. In contrast, PhysDrape focuses on a static body pose, without considering body motion, and its optimization objective differs from that of dynamic methods, preventing a fair comparison.

Qualitative Comparison. Fig. 3 demonstrates the visualization superiority of PhysDrape over DrapeNet. We present results across different body poses and garment types. As highlighted in red boxes, PhysDrape exhibits less interpenetration, more natural draping, fewer unnatural folds, and a more precise fit. Specifically, in areas with body curvature, PhysDrape produces garments that

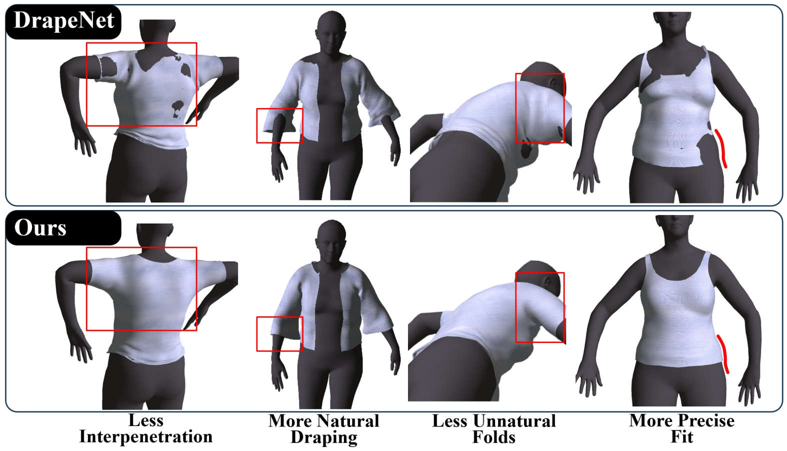

Fig. 3: Garment draping visualization results demonstrating the superiority of PhysDrape over previous state-of-the-art, reflected in less interpenetration, more natural draping, fewer unnatural folds, and a more precise fit.

conform more naturally to the body's contours, appearing more physically realistic. Additionally, at boundary regions, such as cuffs, collars, and underarms, most existing methods suffer from interpenetration, where the hands, neck, or shoulders penetrate the garment model. This issue arises primarily due to the lack of explicit physical constraints, as relying solely on neural network optimization makes it difficult to achieve accurate garment draping results. In contrast, PhysDrape explicitly integrates physical solvers with deep learning models, ensuring more physically plausible garment draping results.

4.3 Ablation Studies

To further demonstrate the effectiveness of our designs in PhysDrape, we conduct a series of ablation experiments, including: 1) the effectiveness of the three modules, 2) the convergence of multi-round simulations in the Stretching Solver, 3) the assessment of runtime acceptability, 4) the effectiveness of joint optimization with the collision handler, 5) the controllability of manually specified material stiffness, and 6) the generalization to different body shapes. More ablation experiments are provided in the supplementary material.

Effectiveness of Modules. Table 2 shows the contribution of the three main modules to the results. As seen in the table, the three modules effectively complement each other, leading to improved performance. Specifically, directly using GNN for regression yields the worst results, as it lacks explicit physical guidance, and the regression results are not physically grounded. The stretching solver is responsible for force-based garment deformation. When used after the

Table 2: Effectiveness of the three contributed modules: Force-Driven GNN, Stretching Solver, and Collision Handler in PhysDrape, measured by strain energy (E_{strain}), bending energy (E_{bend}), and interpenetration ratios (B2G) on the CLOTH3D dataset.

Force-Driven GNN	Stretching Solver	Collision Handler	$E_{\text{strain}} \downarrow$	$E_{\text{bend}} \downarrow$	B2G % ↓
✓			0.20	0.004	3.50
✓	✓		0.17	**0.003**	2.29
✓		✓	0.21	0.004	0.06
✓	✓	✓	**0.15**	0.004	**0.05**

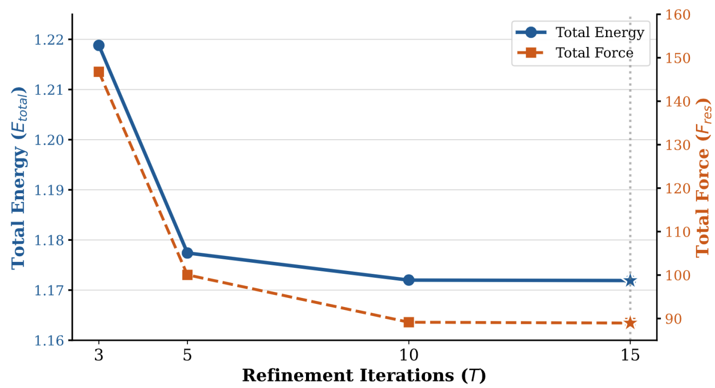

Fig. 4: Total energy (blue line) and total force (orange line) converge as the stretching solver iterates, indicating the effectiveness of the physical solver. This demonstrates that the excess energy in the garment decreases as deformation progresses, and the system converges to a stable value without collapsing.

force-driven GNN, it reduces the internal energy of the fabric through stretching, making the result more plausible. However, without a collision avoidance mechanism, it still leads to a significant increase in the interpenetration ratio, especially within the self-supervised framework. The collision handler significantly reduces the interpenetration ratio by integrating physical constraints into the differentiable model in PhysDrape, achieving approximately a 95% reduction in interpenetration.

Impact of Stretching Solver Iterations. Fig. 4 shows that total energy and residual forces converge to a stable value as the stretching solver iterates. This indicates that, on one hand, the model effectively reduces unnatural stretching, and on the other hand, the reasonable model design prevents structural collapse after multiple force-based stretching iterations.

	Post-Col	Ours
E_{strain}	0.29	**0.15**
E_{bend}	0.005	**0.004**

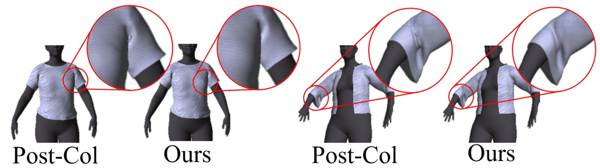

Fig. 5: Compared to using independent collision avoidance postprocessing (Post-Col), our PhysDrape unifies the collision handler within a learnable model, resulting in improved detail accuracy and energy.

Runtime Assessment. The runtime for draping each garment is 29 ms when $T = 3$ and 91 ms when $T = 15$ on a single NVIDIA V100 GPU. These results are obtained experimentally by averaging over 5000 garment draping samples. According to classical user perception theory, 100 ms is the threshold for human perception [25, 28], with responses below this time considered instantaneous. Therefore, whether at $T = 3$ or $T = 15$, PhysDrape achieves state-of-the-art results, as shown in Table 1, and can be regarded as a real-time system.

Impact of Optimizable Collision Handler. Fig. 5 illustrates that integrating collision handling into the optimization process of PhysDrape improves both the physical realism of garment details and reduces overall energy loss, compared to relying on independent postprocessing methods.

Controllability of Material Stiffness. Fig. 6 demonstrates that PhysDrape allows the adjustment of fabric stiffness in garment draping by changing physical parameters. This level of control is difficult to achieve in other deep learning-based garment draping methods, highlighting the controllability and physical realism of the PhysDrape design. Specifically, stiffness can be controlled by adjusting k_{bend} in Eq. 2 and the Lamé constants μ and λ of $\mathbf{P}_T$ in Eq. 1, with higher values resulting in stiffer fabrics. The results show that softer fabrics tend to sag more and exhibit more wrinkling at the crease areas, which aligns with physical observations. This illustrates that PhysDrape not only achieves better quantitative metrics through deep learning generalization but also incorporates physical solvers to generate more physically realistic details with strong controllability.

Generalization to Different Body Shapes Fig. 7 shows that PhysDrape generalizes well across diverse human body shapes. As demonstrated, when applying the exact same garment template to bodies ranging from slightly overweight to slim, our framework seamlessly adapts the fabric to the varying underlying geometries. By explicitly resolving internal active forces and enforcing strict body-garment collision constraints, PhysDrape ensures that the draping remains natural and physically plausible across all proportions.

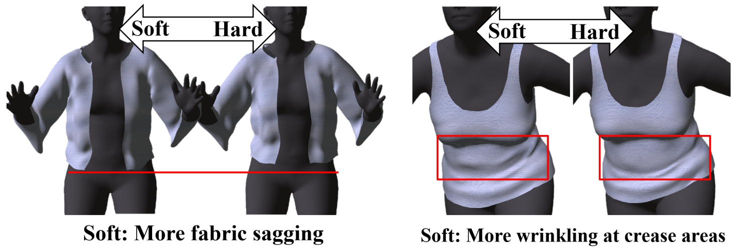

Fig. 6: Visualization results show that material stiffness can be adjusted by manually changing material parameters. Softer materials exhibit more fabric sagging and more wrinkling, particularly at crease areas, consistent with observed physical behavior.

Fig. 7: Comparison of the same garment on different body shapes, ordered from slightly overweight (left) to slim (right).

5 Conclusion

In this work, we present PhysDrape, a novel garment draping framework that bridges physical modeling with deep learning to achieve physically realistic 3D garment deformation. The method consists of three learnable core modules: the force-driven GNN for predicting forces on each node, the stretching solver for modeling fabric deformation, and the collision handler for penalizing interpenetration. The entire framework supports self-supervised, end-to-end optimization and generalizes directly to unseen garment templates. Extensive experiments on the CLOTH3D dataset demonstrate that PhysDrape outperforms previous state-of-the-art methods both quantitatively and qualitatively, while maintaining real-time inference. Ablation studies validate the effectiveness of each design, and visualization results further confirm the physical realism of garment draping as well as the controllable simulation of material stiffness.

Acknowledgements

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province, China, under Grants 2023B0303030003 and 2023B0303040002.

References

1. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 767–778 (2023)
2. Bertiche, H., Madadi, M., Escalera, S.: Cloth3d: clothed 3d humans. In: European Conference on Computer Vision. pp. 344–359. Springer (2020)
3. Bertiche, H., Madadi, M., Escalera, S.: Pbns: physically based neural simulator for unsupervised garment pose space deformation. arXiv preprint arXiv:2012.11310 (2020)
4. Bertiche, H., Madadi, M., Tylson, E., Escalera, S.: Deepsd: Automatic deep skinning and pose space deformation for 3d garment animation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5471–5480 (2021)
5. Bhatnagar, B.L., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-garment net: Learning to dress 3d people from images. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5420–5430 (2019)
6. Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective dynamics: Fusing constraint projections for fast simulation. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 787–797 (2023)
7. Chen, R., Chen, L., Parashar, S.: Gaps: Geometry-aware, physics-based, self-supervised neural garment draping. In: 2024 International Conference on 3D Vision (3DV). pp. 116–125. IEEE (2024)
8. Chibane, J., Pons-Moll, G., et al.: Neural unsigned distance fields for implicit function learning. Advances in Neural Information Processing Systems **33**, 21638–21652 (2020)
9. De Luigi, L., Li, R., Guillard, B., Salzmann, M., Fua, P.: Drapenet: Garment generation and self-supervised draping. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1451–1460 (2023)
10. Designer, M.: `https://www.marvelousdesigner.com/` (2026)
11. Etzmuss, O., Gross, J., Strasser, W.: Deriving a particle system from continuum mechanics for the animation of deformable objects. IEEE Transactions on Visualization and Computer Graphics **9**(4), 538–550 (2003)
12. Grigorev, A., Black, M.J., Hilliges, O.: Hood: Hierarchical graphs for generalized modelling of clothing dynamics. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16965–16974 (2023)
13. Gundogdu, E., Constantin, V., Parashar, S., Seifoddini, A., Dang, M., Salzmann, M., Fua, P.: Garnet++: Improving fast and accurate static 3d cloth draping by curvature loss. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(1), 181–195 (2020)
14. Gundogdu, E., Constantin, V., Seifoddini, A., Dang, M., Salzmann, M., Fua, P.: Garnet: A two-stream network for fast and accurate 3d cloth draping. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8739–8748 (2019)
15. Islam, T., Miron, A., Liu, X., Li, Y.: Deep learning in virtual try-on: A comprehensive survey. IEEE Access **12**, 29475–29502 (2024)
16. Lewiner, T., Lopes, H., Vieira, A.W., Tavares, G.: Efficient implementation of marching cubes' cases with topological guarantees. Journal of graphics tools **8**(2), 1–15 (2003)
17. Li, M., Ferguson, Z., Schneider, T., Langlois, T.R., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Trans. Graph. **39**(4), 49 (2020)

18. Li, R., Guillard, B., Fua, P.: Isp: Multi-layered garment draping with implicit sewing patterns. Advances in Neural Information Processing Systems **36**, 40294–40319 (2023)
19. Li, R., Guillard, B., Remelli, E., Fua, P.: Dig: Draping implicit garment over the human body. In: Proceedings of the Asian conference on computer vision. pp. 2780–2795 (2022)
20. Liao, Z., Wang, S., Komura, T.: Senc: Handling self-collision in neural cloth simulation. In: European Conference on Computer Vision. pp. 385–402. Springer (2024)
21. Liu, T., Bouaziz, S., Kavan, L.: Quasi-newton methods for real-time simulation of hyperelastic materials. Acm Transactions on Graphics (TOG) **36**(3), 1–16 (2017)
22. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866 (2023)
23. Macklin, M., Müller, M., Chentanez, N.: Xpbd: position-based simulation of compliant constrained dynamics. In: Proceedings of the 9th International Conference on Motion in Games. pp. 49–54 (2016)
24. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: Amass: Archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5442–5451 (2019)
25. Miller, R.B.: Response time in man-computer conversational transactions. In: Proceedings of the December 9-11, 1968, fall joint computer conference, part I. pp. 267–277 (1968)
26. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. Journal of Visual Communication and Image Representation **18**(2), 109–118 (2007)
27. Narain, R., Samii, A., O'brien, J.F.: Adaptive anisotropic remeshing for cloth simulation. ACM transactions on graphics (TOG) **31**(6), 1–10 (2012)
28. Nielsen, J.: Usability engineering. Morgan Kaufmann (1994)
29. Nvidia: Nvcloth (2018)
30. Nvidia: Nvidia flex. `https://developer.nvidia.com/flex` (2018)
31. Optitex: Optitex fashion design software. `https://optitex.com/` (2018)
32. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019)
33. Patel, C., Liao, Z., Pons-Moll, G.: Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7365–7375 (2020)
34. Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P.: Learning mesh-based simulation with graph networks. In: International conference on learning representations (2020)
35. Provot, X.: Collision and self-collision handling in cloth model dedicated to design garments. In: Computer Animation and Simulation'97: Proceedings of the Eurographics Workshop in Budapest, Hungary, September 2–3, 1997. pp. 177–189. Springer (1997)
36. Provot, X., et al.: Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In: Graphics interface. pp. 147–147. Canadian Information Processing Society (1995)
37. Santesteban, I., Otaduy, M.A., Casas, D.: Learning-based animation of clothing for virtual try-on. In: Computer Graphics Forum. vol. 38, pp. 355–366. Wiley Online Library (2019)

38. Santesteban, I., Otaduy, M.A., Casas, D.: Snug: Self-supervised neural dynamic garments. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8140–8150 (2022)
39. Santesteban, I., Thuerey, N., Otaduy, M.A., Casas, D.: Self-supervised collision handling via generative 3d garment models for virtual try-on. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11763–11773 (2021)
40. Shi, W., Wong, W., Zou, X.: Generative ai in fashion: Overview. ACM Transactions on Intelligent Systems and Technology **16**(4), 1–73 (2025)
41. Stuyck, T., Chen, H.y.: Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics. Proceedings of the ACM on Computer Graphics and Interactive Techniques **6**(3), 1–14 (2023)
42. Su, T., Zhang, Y., Zhou, Y., Yu, Y., Du, S.: Gpu-based real-time cloth simulation for virtual try-on. In: PG (Short Papers and Posters). pp. 1–2 (2018)
43. Tan, Q., Zhou, Y., Wang, T., Ceylan, D., Sun, X., Manocha, D.: A repulsive force unit for garment collision handling in neural networks. In: European conference on computer vision. pp. 451–467. Springer (2022)
44. Tang, M., Yuan, R., Kou, D., Sun, M., Zhang, L.: Saf: Local shape-aware face-based garment collision handling via neural sdfs. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
45. Tiwari, L., Bhowmick, B., Sinha, S.: Gensim: Unsupervised generic garment simulator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4169–4178 (2023)
46. Vassilev, T., Spanlang, B., Chrysanthou, Y.: Fast cloth animation on walking avatars. In: Computer Graphics Forum. vol. 20, pp. 260–267. Wiley Online Library (2001)
47. Venkatesh, R., Karmali, T., Sharma, S., Ghosh, A., Babu, R.V., Jeni, L.A., Singh, M.: Deep implicit surface point prediction networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12653–12662 (2021)
48. Xi, N., Hamari, J.: Shopping in virtual reality: A literature review and future agenda. Journal of Business Research **134**, 37–58 (2021)
49. Zeller, C.: Cloth simulation on the gpu. In: ACM SIGGRAPH 2005 Sketches, pp. 39–es (2005)
50. Zhang, Z.: Improved adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS). pp. 1–2. Ieee (2018)
51. Zhao, F., Wang, W., Liao, S., Shao, L.: Learning anchored unsigned distance functions with gradient direction alignment for single-view garment reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12674–12683 (2021)