Gaussian Garments: Reconstructing Simulation-Ready Clothing with Photorealistic Appearance from Multi-View Video

Boxiang Rong[*1] Artur Grigorev[*1,2] Wenbo Wang[1] Michael J. Black[2]
Bernhard Thomaszewski[1] Christina Tsalicoglou[1] Otmar Hilliges[1]

[1]Department of Computer Science, ETH Zurich
[2]Max Planck Institute for Intelligent Systems, Tübingen

https://ribosome-rbx.github.io/Gaussian-Garments

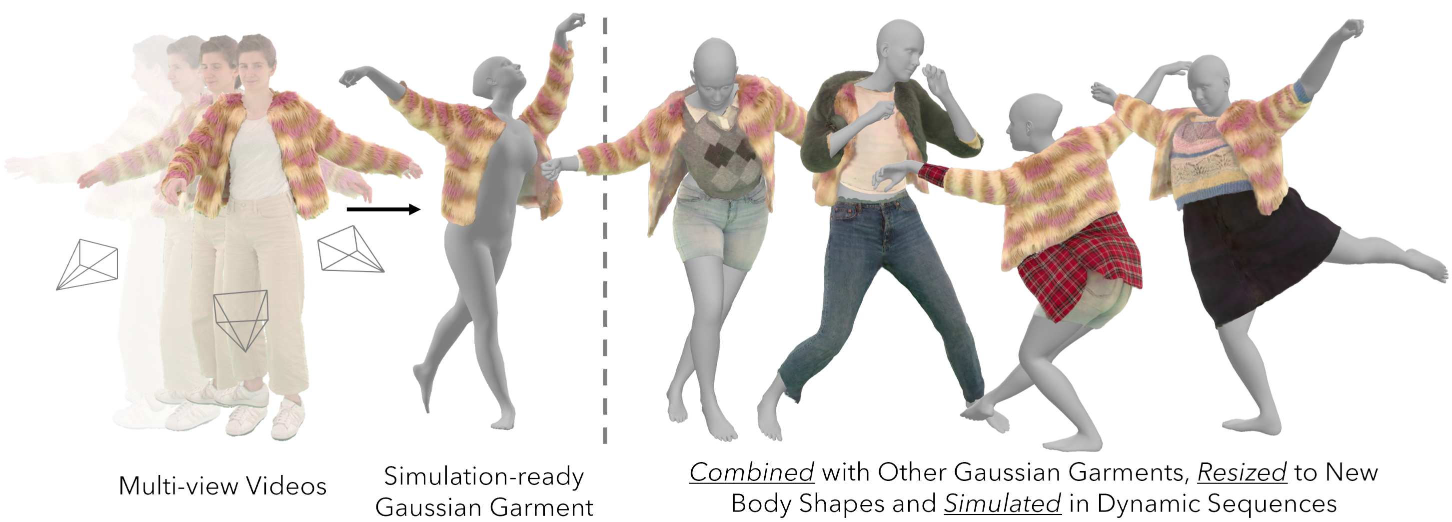

Figure 1. We introduce Gaussian Garments, a novel approach for reconstructing realistic simulation-ready garments from multi-view videos. Our natural coupling of 3D meshes and 3D Gaussian splatting allows *Gaussian Garments* to accurately represent both the overall geometry and the high-frequency details of human clothing. The reconstructed garments can then be retargeted to novel human models, resized to fit novel body shapes, and simulated over moving bodies with novel motions. Our approach also enables the automatic construction of complex multi-layer outfits from a set of separately captured Gaussian garments.

Abstract

We introduce Gaussian Garments, a novel approach for reconstructing realistic simulation-ready garment assets from multi-view videos. Our method represents garments with a combination of a 3D mesh and a Gaussian texture that encodes both the color and high-frequency surface details. This representation enables accurate registration of garment geometries to multi-view videos and helps disentangle albedo textures from lighting effects. Furthermore, we demonstrate how a pre-trained graph neural network (GNN) can be fine-tuned to replicate the real behavior of each garment. The reconstructed Gaussian Garments can be automatically combined into multi-garment outfits and animated with the fine-tuned GNN.

*Authors contributed equally

1. Introduction

Reconstructing and animating human apparel is essential for many applications, from virtual try-on systems to movies and video games.

Faithful digital representation of real garments requires capturing three key aspects. First, the 3D *geometry* of the garments must be reconstructed to model both their overall structure and fine details. Second, the *appearance* of the garments must be recreated to accurately reflect their color and texture. Finally, the real *behavior* of the garments must be mimicked to produce convincing animations. Our method, Gaussian Garments, leverages the expressivity of 3D Gaussian splatting to reconstruct these three critical aspects from multi-view videos.

In computer graphics, garments are traditionally represented as polygonal meshes with 2D textures. While this representation enables efficient simulation and appealing rendering, creating detailed garment meshes is labor-intensive, particularly for complex textures like fur. Further, meshes are not well-suited for differentiable optimization of their structure and topology from images. To overcome these limitations, recent works have started to explore neural implicit representations (NIRs) as a basis for modeling photorealistic clothing. While NIRs provide strong flexibility in terms of clothing topology and appearance, using them to generate physically realistic motions is exceedingly difficult.

3D Gaussian splatting has recently emerged as a highly efficient and flexible alternative for photorealistic scene reconstruction. Unlike NIRs, Gaussians can be edited individually to accommodate changes in scene geometry, appearance, and lighting. Recent works leverage this ability to generate photorealistic digital copies of clothed humans. However, these methods construct holistic avatars without the ability to extract individual garments as separate assets. Consequently, they cannot retarget these garments to different bodies, adjust their size, or combine clothing items from various avatars into a novel outfit—tasks crucial for many computer graphics applications.

In this work, we introduce Gaussian Garments—the first method that uses 3D Gaussian splatting to reconstruct photorealistic, simulation-ready assets of human clothing. At its core, our method combines mesh-based geometry with Gaussian-based appearance modeling. Starting with an initial garment mesh obtained from multi-view images we register it to a set of multi-view videos using a photometric optimization procedure based on Gaussian splatting. Then, we optimize a Gaussian texture to recover the garment's detailed appearance, with disentangled ambient color and view-dependent properties. Finally, using the registered mesh, we fine-tune a graph neural network (GNN) for neural simulation to match the garment's real-world behavior.

In summary, our main contributions are

- a comprehensive pipeline for reconstructing the shape, appearance, and behavior of real-world garments using Gaussian splatting,
- an algorithm for registering garment meshes to multi-view videos with an optimization procedure based on Gaussian splatting, and
- a Gaussian Garment representation that combines triangle meshes with Gaussian textures to capture photorealistic appearance and can be used as a fully controllable 3D asset.

2. Related work

2.1. Garment reconstruction

Reconstructing 3D representations of real-world garments is a long-studied task. Input data in this problem ranges from 4D scans and multi-view videos to single images.

ClothCap [24] and SIZER [34] segment 4D scans to extract garment meshes, which can be retargeted to novel body shapes and poses. CaPhy [33] further optimizes a neural network to animate garments while preserving physical properties. Bang et al. [1], NeuralTailor [13] and SewFormer [19] estimate sewing patterns from static 3D scans, point clouds and monocular images, respectively. These sewing patterns can then be draped over body geometry to produce a 3D mesh. DiffAvatar [15] employs static 3D representation to jointly optimize both the garment's 2D pattern and material properties, resulting in simulation-ready meshes.

BCNet [8] and SMPLicit [4] train neural networks to generate template-mesh displacements and unsigned neural fields, respectively, from monocular images. DeepFashion3D [43], Zhu et al. [44] and SelfRecon [9] use joint explicit–implicit representations to register garment templates to 2D images. REC-MV [27] reconstruct temporally consistent surfaces from monocular videos. However, these methods do not reconstruct the garment's appearance.

SCARF [5] uses monocular videos to optimize an articulated neural radiance field (NeRF). While it can model garments' appearance over novel body shapes and poses, it suffers from the choice of representation. NeRF reconstructions produce poor geometries and are affected by slow optimization and rendering speed. Additionally, SCARF does not allow combining different reconstructed garments.

Closest to our approach are the works by Xiang et al. [37, 38]. They reconstruct textured garment meshes from multi-view videos, with [37] also using a physical simulator to generate cloth dynamics. While achieving high visual quality, they use simple textured meshes to represent garments, which limits their ability to model high-frequency geometric details like fur. Moreover, they do not provide a means to reconstruct material parameters for the garments and select them manually instead.

With Gaussian Garments, we demonstrate how 3D garment meshes can be combined with 3D Gaussian splatting technique to achieve photorealistic garment appearance. Additionally, we fine-tune a garment-modeling GNN to accurately replicate the real garment behavior.

2.2. 3D Gaussian splatting for human avatars

The recently proposed 3D Gaussian splatting (3DGS) technique [11] reconstructs scenes using explicitly defined 3D Gaussian kernels. This method combines the advantages of both implicit and explicit 3D representations. Similar to Neural Radiance Fields (NeRFs) [2, 32] and neural signed distance fields [22, 35], the 3DGS representation

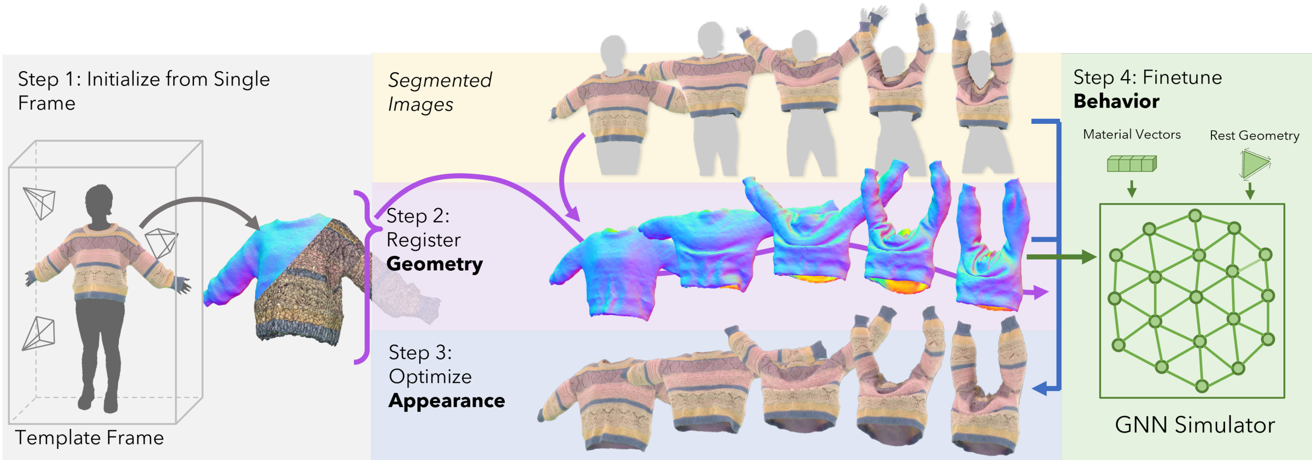

Figure 2. The procedure for obtaining simulation-ready photorealistic garment assets consists of four steps. In Step 1, we initialize the garment's geometry and appearance from a single multi-view frame (Sec. 3.1). In Step 2, we register the garment geometry to multi-view videos (Sec. 3.2). In Step 3, we optimize the garment's appearance over the training sequences (Sec. 3.3). In Step 4, we fine-tune a simulation GNN to accurately replicate the garment's real behavior (Sec. 3.5). The resulting garment assets can be directly simulated with the GNN, combined into multi-garment outfits, and resized to fit different body shapes.

can be optimized from multi-view images and is capable of flexibly modeling diverse topologies. Additionally, the explicit nature of 3DGS enables it to easily represent dynamic scenes [20] and model physical behavior [39].

Despite its recent introduction, 3D Gaussian splatting has been adopted by numerous approaches to represent humans in digital environments. GaussianAvatars [25] and SplattingAvatar [31] use parametric meshes with rigidly attached 3D Gaussians to represent human heads and clothed bodies. GaussianAvatar [7] and 3DGS-Avatar [26] optimize a canonical Gaussian body, a skinning model, and a neural network that predicts pose-dependent offsets to the Gaussian parameters. AnimatableGaussians [16] construct a person-specific canonical template and predict a Gaussian texture containing appearance and geometry parameters. The canonical template and diffused skinning model allow [16] to better model loose garments. PhysAvatar [41] uses 3D Gaussian splatting to register meshes of clothed humans to multi-view videos. It then uses inverse rendering to reconstruct the mesh textures and inverse physics to recover material parameters. For its final representation, PhysAvatar discards 3D Gaussians and uses flat-textured meshes instead. This enables relighting the meshes with standard techniques but does not allow the modeling of non-flat surfaces like fur. Moreover, PhysAvatar does not provide a means to reconstruct template meshes for clothed humans and uses ground-truth ones instead.

A common drawback of these methods is their focus on reconstructing holistic avatars of clothed humans without separating garments from the bodies. This limitation reduces their applicability in common computer graphics tasks such as simulating garments over different human models, combining garments into outfits, and fitting garment sizes to varying body shapes. D3GA [45] and LayGA [18] address this issue by separating garments from human bodies, but are limited to modeling simple tight-fitting outfits consisting of two garments (e.g. a T-shirt and pants).

In contrast, Gaussian Garments reconstructs distinct 3D garment assets that can be resized and combined into multi-layer outfits. Fine-tuning a cloth simulation GNN enables realistic modeling of loose garments in dynamic motions.

3. Method

We use a set of multi-view videos to reconstruct geometry, appearance, and behavior of a real-world garment. Our pipeline, outlined in Fig. 2, consists of four main stages. First, we initialize Gaussian garment geometry and appearance from a single multi-view frame (Sec. 3.1). Second, we register the garment's geometry to all available frames (Sec. 3.2). Third, we optimize the garment's appearance by disentangling the albedo Gaussian texture from lighting effects and per-frame local transformation offsets predicted by a neural network (Sec. 3.3). Finally, we fine-tune the garment's behavior by optimizing a graph neural network (GNN) [6] to replicate the registered garment motion (Sec. 3.5). In this section, we detail each of these steps.

Note that apart from RGB frames, our pipeline requires 2D semantic segmentation maps and parametric human body models [23] fitted to each frame. We extract this information from multi-view videos automatically using existing method [36].

3.1. Gaussian garment initialization

3.1.1. Mesh reconstruction

As an initial step, we reconstruct the static geometry of a given garment. For that, we select a "template" multi-view frame where the garment's is surface fully visible. We recover the garment's 3D mesh from this frame, using existing algorithms for multi-view stereo [30], surface reconstruction [10], and remeshing [14] (see Sec.1 in Supp. Mat.).

Together with the Gaussian texture, described below, the meshes obtained in this step can represent both the overall garment geometry and high-frequency details like fur.

3.1.2. Gaussian texture

To represent the garment's appearance, we use a so-called Gaussian texture. Similar to a traditional texture, it maps between the 3D mesh surface and a 2D texture image that controls the surface appearance. However, in our case, each point on the texture defines parameters for a 3D Gaussian: spherical harmonic coefficients $\boldsymbol{\phi} \in [0,1]^{16\times 3}$, opacity α, scale $\mathbf{s} \in \mathbb{R}^3_+$, local rotation $\mathbf{r} \in \mathbb{H}$ and translational offsets $\boldsymbol{\mu} \in \mathbb{R}^3$. The latter two are set in a local coordinate frame which we define later.

We use the Gaussian texture and the mesh geometry to construct a Gaussian garment in 3D space in the following way. We first sample the Gaussians from the texture in a regular grid (e.g., once per texel). The Gaussian's location on the texture controls which 3D face f_i it is attached to and what its barycentric coordinates within f_i are. These two elements define the initial position of the Gaussian on the mesh surface. We call this position the Gaussian's "surface point". This surface point serves as the origin for the Gaussian's local coordinate frame. The basis of this coordinate frame consists of the normal vector for the face f_i and two orthogonal vectors on its surface (see Fig. 3, left).

Following Qian et al. [25], we determine the Gaussian's final 3D position and shape using its scale $\mathbf{s}$, rotation quaternion $\mathbf{r}$, and translational offsets $\boldsymbol{\mu}$. See Sec.2 of the Supp. Mat. for details.

3.2. Tracking-based registration

To use Gaussian Splatting for geometry registration, we first have to construct an initial appearance model represented by 3D Gaussians. To do so, we initialize the Gaussian texture with default parameters, create Gaussians on the mesh surface, and optimize them to match the template-frame observations (see Sec. 2.1 in Supp. Mat. for details). This initial appearance model is only used for mesh registration. We enhance its visual quality and disentangle albedo color from lighting effects in later steps (Sec. 3.3). After obtaining a template garment mesh and an initial appearance model, we register the template mesh to multi-view videos. The key to this process is propagating the gradient from the image space to the positions of the mesh nodes. To achieve

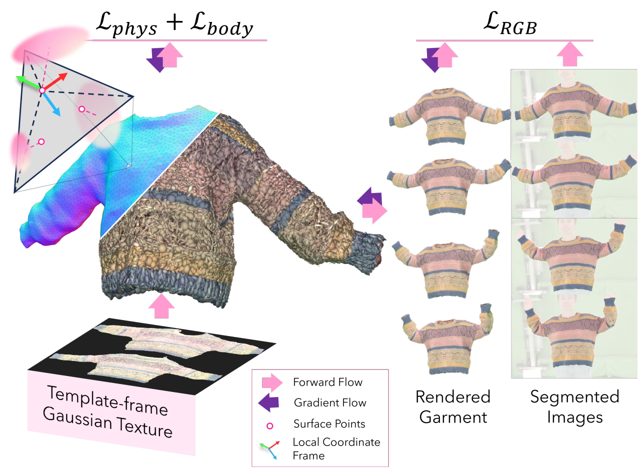

Figure 3. To register the garment mesh we render the Gaussians rigidly attached to the mesh faces (top left) and optimize a combination of the RGB loss $\mathcal{L}_{RGB}$ and physical energies $\mathcal{L}_{phys}$. We also use a body penetration term $\mathcal{L}_{body}$ to ensure that the garment conforms to the body model.

this, we compute the error $\mathcal{L}_{RGB}$ between the rendered Gaussian splats and the ground-truth images. We then pass its gradients through the 3D Gaussians, rigidly attached to the garment's faces, to the nodes of the garment mesh. $\mathcal{L}_{RGB}$ is defined as

$$\mathcal{L}_{RGB} = \lambda_{RGB}\mathcal{L}_1 + (1-\lambda_{RGB})\mathcal{L}_{SSIM}, \tag{1}$$

where $\mathcal{L}_1$ is a mean absolute error, $\mathcal{L}_{SSIM}$ is a structural similarity loss, and λ_{RGB} is a balancing weight.

However, naïve minimization of the RGB discrepancy $\mathcal{L}_{RGB}$ between renders and observations would result in severely disfigured meshes (see Fig.1 in the Supp. Mat.). Therefore, we expand the optimized loss function with a set of physical energies.

First, we regularize the angle between each pair of neighboring faces with bending energy $\mathcal{L}_{bending}$:

$$\mathcal{L}_{bending} = \sum_{(i,j)} \frac{\|e_{ij}\|^2}{a_{ij}} \operatorname{atan2}(\sin(\theta_{ij}), \cos(\theta_{ij}))^2, \tag{2}$$

where (i,j) are indices of neighboring triangles, θ_{ij} is the angle between the triangles' normal vectors, $\|e_{ij}\|$ is the length of the edge connecting the two triangles, and a_{ij} is the sum of their areas.

Second, we regularize the stretching of the triangles relative to the template frame using the strain energy $\mathcal{L}_{strain}$, based on the St. Venant–Kirchhoff material model. This energy uses the deformation gradient $\mathbf{F} = \dfrac{\partial x_t}{\partial X}$ of the current frame geometry x_t relative to the template geometry X, and is computed as a sum over all faces f_i:

$$\mathcal{L}_{strain} = \sum_i V_i \left(\frac{\lambda}{2} \operatorname{tr}(\mathbf{G}_i)^2 + \mu \operatorname{tr}(\mathbf{G}_i^2) \right). \tag{3}$$

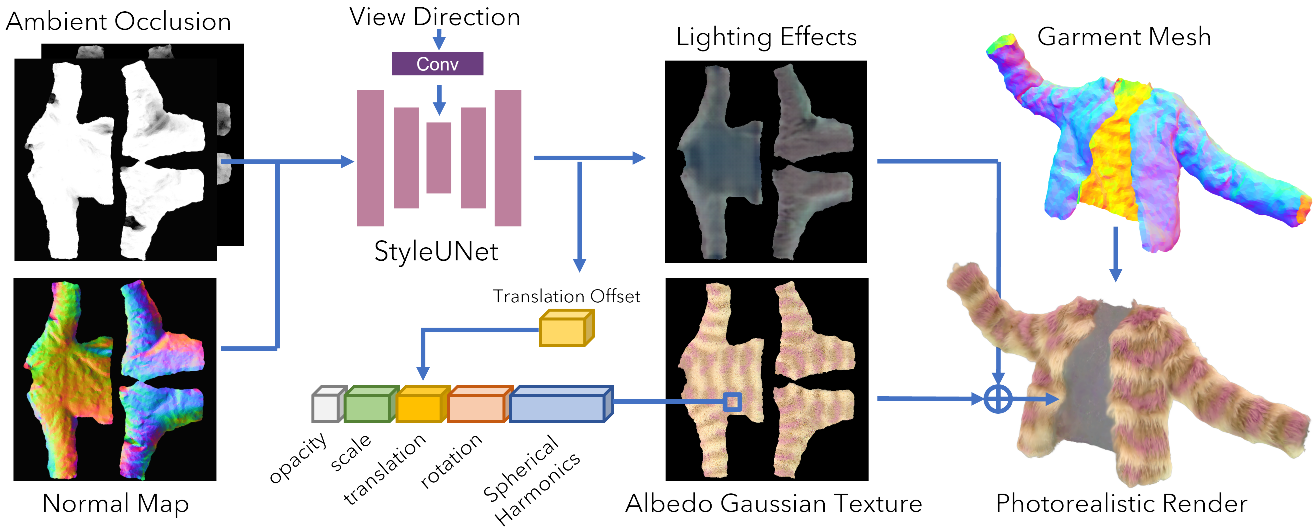

Figure 4. We model the appearance of Gaussian Garments using a combination of an albedo Gaussian texture and a neural network that predicts lighting effects and local translational offsets. The albedo Gaussian texture stores color information along with Gaussian parameters, including local rotation, translation, and scale. During rendering we regularly sample the Gaussian texture and spawn the 3D Gaussians rigidly attached to the garment surface.

Here, $\mathbf{G_i}$ is the Green strain tensor for the face f_i: $\mathbf{G_i} = \frac{1}{2}(\mathbf{F_i}^T\mathbf{F_i} - \mathbf{I})$, V_i is the face's volume (thickness$\times$area), and λ and μ are Lamé coefficients serving as balancing weights. For λ and μ we use the same default values as in SNUG [29].

We denote the full physical-regularization term as $\mathcal{L}_{phys} = \mathcal{L}_{bending} + \mathcal{L}_{strain}$. It helps preserve the physical realism of the tracked mesh but does not provide any information about the underlying body. Hence, the garments tend to implode and not conform to the body shape. This issue can be solved using a parametric body mesh fitted to the multi-view sequence. Following ContourCraft [6], we use cubic energy term to penalize negative normal distance between garment nodes and the body faces closest to them:

$$\mathcal{L}_{body} = \sum_i \max(\epsilon_{body} - ((v_i - f_i) \cdot \vec{n}_i), 0)^3, \quad (4)$$

where v_i are the vertex coordinates, f_i is a point on the body face, $\vec{n}_i$ is this face's normal vector, and ϵ_{body} is a safety margin. In our experiments, we set ϵ_{body} to 3mm.

However, for sequences with dynamic body motions, the optimization process often starts far from the target body pose. This large difference in vertex positions causes the optimization to produce unrealistic geometries or diverge completely (see Fig.1 in Supp. Mat. for illustrations). We work around this issue by substituting $\mathcal{L}_{body}$ with a simple ersatz regularization, $\mathcal{L}_{VE}$, in the first half of the optimization process. This regularization uses virtual edges built between the garment faces opposite each other in the template-frame geometry and penalizes these face pairs for getting too close together (see Sec.3.1 in Supp.Mat. for details). In the second half of the optimization, after the RGB signal pulls the garment geometry to better conform to the body pose, we switch to using the body penetration term $\mathcal{L}_{body}$.

The full energy term minimized in the registration process is formulated as

$$\mathcal{L}_{register} = \lambda_1\mathcal{L}_{RGB} + \lambda_2\mathcal{L}_{phys} + \lambda_3\mathcal{L}_{body}, \quad (5)$$

with $\mathcal{L}_{body}$ here substituted by $\mathcal{L}_{VE}$ for the first half of the optimization process in each frame.

3.3. Appearance reconstruction

So far, we have used the Gaussian texture reconstructed from the template frame. While it provides useful gradients for the registration procedure, its quality is limited by the visual information available in a single time frame. Moreover, the lighting conditions are baked into this texture. Therefore, we further optimize the garment's appearance using multi-view videos and meshes registered in the previous step.

We disentangle the garment's appearance into two components: a) a base Gaussian texture, introduced in Section 3.1.2. b) a texture update predicted by a neural network f_θ. This neural network takes as input the albedo occlusion map A and the normal map N of the mesh. Following Li et al. [16], we choose the StyleUNet architecture for f_θ. It predicts offsets to the texture's spherical harmonics $\Delta\phi$, and translations $\Delta\boldsymbol{\mu}$ in each frame.

The predicted offsets to the spherical harmonic coefficients allow the model to separate the albedo colors stored in the base texture from lighting effects (see Fig. 5). The translational offsets account for observational noise, preserving high-frequency detail and local geometry of the surface (see Fig.3 in Supp. Mat. for visual examples).

Figure 5. We disentangle the albedo color of the Gaussian Garments from the lighting effects predicted by a neural network. Here we show four garments rendered over the registered sequence. Note how, when rendered with albedo colors, the garments lack any shadows or specular effects. The lighting information comes solely from network predictions and matches the ground-truth information. The figure shows registered mesh sequences that were not seen by the appearance model during training.

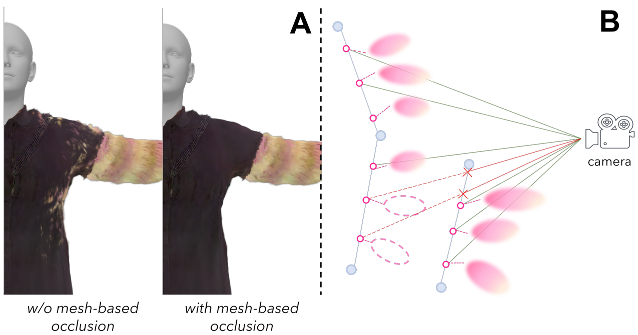

Figure 6. When a "fuzzy" garment is placed under another one, its Gaussians phase through the outer garment (A, left). We solve this by checking the visibility of the Gaussians' surface points based on the mesh geometries (B). Only the Gaussians with visible surface points are rendered (A, right).

The final Gaussian texture Ω for a specific frame is formulated as follows:

$$\Omega = \{\boldsymbol{\phi} + \Delta\boldsymbol{\phi}, \alpha, \mathbf{s}, \mathbf{r}, \boldsymbol{\mu} + \Delta\boldsymbol{\mu}\} \in \mathbb{R}^{H\times W\times 59}, \quad (6)$$

where $\Delta\boldsymbol{\phi}$ and $\Delta\boldsymbol{\mu}$ are predicted by f_θ:

$$\Delta\boldsymbol{\phi}, \Delta\boldsymbol{\mu} = f_\theta(A, N). \quad (7)$$

3.4. Mesh-based 3DGS rendering

When modeling surfaces in close proximity using 3D Gaussian splatting, it is crucial to properly handle the visibility of the Gaussians. For instance, if a fuzzy surface (e.g., fur) is placed beneath an outer garment layer, the inner layer's Gaussians would incorrectly phase through it (see Fig. 6A), whereas in reality, the fur would be pressed down by the outer layer. Properly modeling effects like this would require simulating physical behavior on a per-Gaussian level. We address this issue with a simple yet effective workaround that leverages the coupling between the mesh and 3D Gaussian splatting representations.

For each Gaussian, we cast a ray from the camera origin to its corresponding surface point, defined by a mesh face and the point's barycentric coordinates. We then check if this point is occluded by another mesh, such as the human body or another garment, and only render the Gaussian if its corresponding surface point is visible (see Fig. 6B).

3.5. Behavior fine-tuning

In the final stage of our pipeline, we optimize the garment's behavior. To simulate the dynamics of Gaussian garments, we employ a learned graph neural network introduced in ContourCraft [6]. This GNN, denoted as g_ψ, where ψ are the network's parameters, takes as input the nodal positions $\mathbf{x}_t$ and velocities $\mathbf{v}_t$ of the mesh at the current frame t, along with each node's material vector $\mathbf{m}$ and each edge's resting geometry $\bar{E}$. From these inputs, g_ψ predicts the nodal accelerations $\hat{\mathbf{a}}_{t+1}$ for the next frame:

$$\hat{\mathbf{a}}_{t+1} = g_\psi(\mathbf{x}_t, \mathbf{v}_t, \mathbf{m}, \bar{E}) \quad (8)$$

To fit the observed behavior of the garment, we jointly optimize the model's weights, the material vectors, and the rest edges to minimize the loss function $\mathcal{L}_{behavior}$. This loss function combines the mean squared error between the predicted and registered nodal positions with a set of physical terms.

$$\begin{aligned}\psi^*, \mathbf{m}^*, \bar{E}^* = \underset{\psi^*, \mathbf{m}^*, \bar{E}^*}{\operatorname{argmin}} \Big[\\ \sum_t \mathcal{L}_{behavior}(g_\psi(\mathbf{x}_t, \mathbf{v}_t, \mathbf{m}, \bar{E}), \mathbf{a}_{t+1}) \Big]\end{aligned} \quad (9)$$

where $\mathbf{a}_{t+1}$ are the nodal accelerations in frame $t+1$ in a registered sequence. For more details, please refer to Sec. 4 of the supplementary material.

4. Results

4.1. Data

In total, we use 15 garments in our experiments, of which 13 are part of the 4D-Dress dataset [36], and two are newly captured garments with fuzzy fur-like textures. The subjects wearing the garments are recorded by 48 cameras regularly placed around them. Each garment is captured in 6 to 10

Table 1. Quantitative ablation study of our registration algorithm.

	F-score, % ↑	CD, cm ↓	p2m, cm ↓	$\mathcal{L}_{body}$ ↓
only-RGB	4.4	15.5e+2	6.19e-1	3.47e+2
w/o body	88.4	1.12	5.24e-1	4e-3
w/ body	89.1	6.57	5.26e-1	4.24e-1
Ours-full	**89.6**	**1.04**	**5.04e-1**	**1.03e-5**

video sequences with diverse poses of roughly 150 frames each. We use 44 cameras to reconstruct, register, and train the appearance models and validate our results using the remaining 4 cameras. We train the appearance model and fine-tune the simulation GNN with all multi-view videos available for the given subject except one, holding it out as a validation set. This way, we evaluate the trained parts of the pipeline (appearance and behavior optimization) on the pose sequences unseen during training. In our qualitative evaluation and supplementary video, we also use sequences from the AMASS dataset [21] demonstrating our ability to generalize to completely new poses and body shapes.

4.2. Garment registration

We evaluate our algorithm for tracking-based mesh registration. In this section, we compare it to several ablations. In Sec.6.1 of the Supp.Mat. we also compare our registration procedure to the state-of-the-art registration method by Lin et al. [17]. We demonstrate that using multi-view videos our method achieves comparable, albeit slightly lower, accuracy to [17], while the latter optimizes template meshes using ground-truth scanned geometries.

For the quantitative analysis, we use three metrics: Chamfer Distance (CD), average point-to-mesh distance (p2m), and F-score [12], which intuitively describes the percentage of correctly reconstructed points on the mesh surface. We use a threshold value of 1cm for the F-score. These metrics measure how close the registration results are to the ground-truth garment meshes, which are reconstructed with a system similar to that used in [3]. To obtain the individual garment parts, we perform semantic segmentation with the method proposed by Wang et al. [36]. We also compute the body penetration loss $\mathcal{L}_{body}$ to measure how well the registered mesh aligns with the underlying body geometry.

The first ablation *"only-RGB"* optimizes the positions of the mesh vertices using only the RGB signal $\mathcal{L}_{RGB}$ without any physics-based regularization. In this case, the optimized mesh completely loses its structure, producing disfigured geometry spatially distant from the ground truth (Table 1). The second ablation adds two physical terms to the optimization energy: $\mathcal{L}_{bending}$ and $\mathcal{L}_{stretching}$. They serve as regularization and help to keep garment geometry physically plausible. However, the garments optimized without the body geometries tend to implode and do not conform to the observed body. In Table. 1 we call this ablation *"w/o body." "w/ body"* uses the body penetration term $\mathcal{L}_{body}$ with respect to the parametric body meshes. This improves the draping in most cases, but in dynamic pose sequences, the optimization may start far away from the next frame body mesh leading to divergence and worse metric values on average. In our full registration pipeline *"Ours-full,"* we use a substitute loss $\mathcal{L}_{VE}$ for the first half of the optimization and then switch back to $\mathcal{L}_{body}$ (Sec. 3.2). This way, we successfully register the dynamic movement of loose garments like dresses and open jackets (see Fig.1 of the supplementary document for qualitative comparison of the ablations and the supplementary video for further result visuals).

Table 2. We quantitatively compare our full appearance model to a set of ablations over the unseen pose sequences and unseen camera views. Predicting lighting effects and per-frame translation offsets allows us to better match the ground-truth observations.

	LPIPS ↓	SSIM ↑	PSNR ↑
template-frame	1.09e-2	0.988	36.5
only-texture	9.52e-3	0.990	37.6
w/ lighting	8.52e-3	0.991	38.3
Ours-full	**8.12e-3**	**0.992**	**38.8**

4.3. Appearance modeling

We evaluate the photorealism of our appearance model both quantitatively (Table 2) and qualitatively (Fig.3 in the Supp.Mat.) by comparing it to several ablations. The quantitative evaluation (Table 2) compares the models in terms of three metrics measuring visual realism: structural similarity (SSIM) [42], learned perceptual similarity (LPIPS) [40], and peak signal-to-noise ratio (PSNR). We perform the comparisons using validation videos not seen in training by any of the models and novel camera views.

We first compare our model to a simple *"template-frame"* procedure, which optimizes the Gaussian scene only for the template frame. This bakes the lighting conditions and any visual artifacts present in the template frame into the garment's appearance. On the other hand, naïvely optimizing the Gaussian texture over multiple videos (*"only-texture"*) averages the lighting and high-frequency details, resulting in blurry textures. The ablation *w/ lighting* optimizes a neural network to predict lighting effects from local information – ambient occlusion and normal maps. This helps disentangle the garment's albedo texture from lighting but still averages high-frequency details. Finally, our full model (*Ours-full*) accounts for the noise in the observations by predicting translational offsets for the Gaussians in each frame, which helps preserve high-frequency information and reduce blur.

We also evaluate our behavior-tuning procedure in Sec.4 of the Supp. Mat.

4.4. Applications

Gaussian Garments create comprehensive representations of real-world garments as distinct 3D assets. This opens the door for many applications sought by 3D designers.

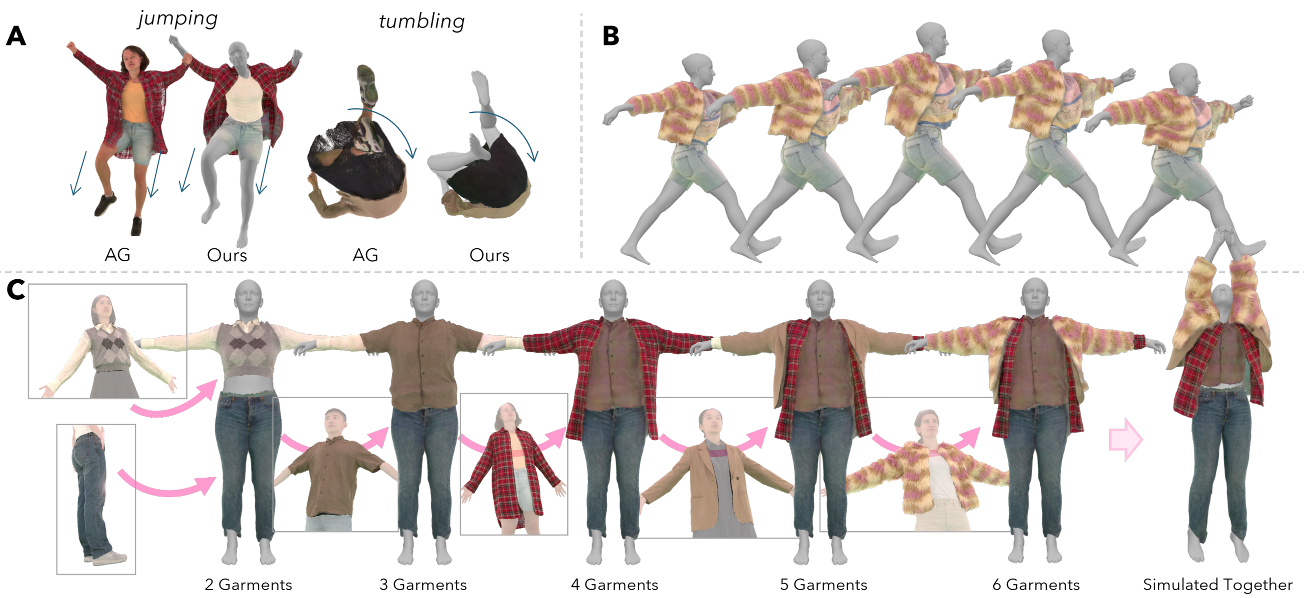

Figure 7. Applications of Gaussian Garments. **A**: Comparison of our method's animations of dynamic pose sequences to Animatable Gaussians [16]. Mesh-based representation and neural simulation make Gaussian Garments much more robust in challenging poses. **B**: We automatically resize the Gaussian Garments to fit the desired body shape. **C**: We automatically combine multiple Gaussian Garments into a single multi-layer outfit. This outfit is then simulated over a new pose sequence (leftmost). All results in this figure show geometries simulated with a fine-tuned GNN over pose sequences from AMASS [21] that were not seen during training.

4.4.1. Simulation

Gaussian garments can be simulated in dynamic sequences by the fine-tuned ContourCraft [6] GNN, which can prevent and resolve cloth self-penetrations, thus automatically modeling re-sized and re-posed multi-layer outfits. The simulation speed ranges from 10 fps for single garments to 1 fps for outfits with multiple layers as in Fig. 7C.

Compared to holistic Gaussian avatars like Animatable Gaussians [16], using our reconstructed garments with the simulation GNN allows us to robustly model challenging pose sequences like jumps and tumbles. In Fig. 7A and in the supplementary video, we demonstrate examples where Gaussian Garments excel in modeling the reconstructed outfits compared to [16]. Since our method does not model the non-covered parts of the human body, we do not compare quantitatively to [16]. In Sec. 6.2 of the Supp. Mat. we also provide a quantitative comparison to SCARF [5].

4.4.2. Mix-and-match

Multiple distinct garment assets, extracted from different multi-view videos, can be combined into novel multi-garment outfits. We first align each individual garment with the canonical pose and shape of the parametric body model SMPL-X [23]. Then, we automatically order the garments by running a simple procedure built around ContourCraft (see Sec. 5 of the Supp. Mat.). The resulting outfit can then be simulated with the fine-tuned ContourCraft model. In Fig. 7C, we show how we automatically combine garments into a single simulation-ready outfit.

4.4.3. Garment resizing

The reconstructed Gaussian garments and their combinations can be automatically re-sized to match a given body shape, by adjusting the edge lengths in the garments' rest geometry according to the shape blend-weights collected from the SMPL-X body. We diffuse the body model's blend weights as proposed by Santesteban et al. [28] to avoid artifacts caused by the resizing. Fig. 7B demonstrates an outfit automatically resized to random body shapes.

5. Limitations and future work

While our method can model the overall geometry and photorealistic appearance of garments, the following limitations are to be addressed in future work. 1) For the appearance model, we assume scenes with uniform lighting. Our approach predicts lighting effects based on ambient occlusion and normal maps but does not accommodate dynamic relighting. 2) While our Gaussian texture can capture high-frequency geometric details like fur to some extent, its effectiveness is limited by the quality of the segmentation masks used during training. 3) We use Gaussian textures with a fixed resolution of 512^2 pixels, which may lead to magnification and minification artifacts. An important direction of future work is adopting standard computer graphics techniques like mipmapping to Gaussian textures. 4) Details such as collars and pockets are represented using the appearance model rather than explicit geometry, as our approach is not aware of the geometry of creases.

Finally, the first three stages of our pipeline can be used with a differentiable physical simulator instead of the learned GNN as long as this simulator allows for optimizing the material parameters of the cloth. We chose ContourCraft [6] for its ability to initialize and recover from self-intersecting geometries and its inference speed.

6. Conclusion

We present Gaussian Garments, a comprehensive approach for creating fully controllable 3D clothing assets from multi-view videos based on 3D Gaussian splatting (3DGS). Our approach seamlessly integrates 3DGS with commonly used polygonal meshes to reconstruct the 3D geometry of garments, register it to video observations, optimize garment appearance to achieve photorealistic quality, and fine-tune garment behavior to model dynamic garment motion. We demonstrate results on garment simulation, mixing-and-matching, and resizing as some of the applications of our Gaussian Garments.

Acknowledgements. AG was supported in part by the Max Planck ETH Center for Learning Systems. We thank Juan Zarate, Wojciech Zielonka, and Peter Kulits for their feedback and help during the project.

References

[1] Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. Estimating garment patterns from static scan data. In *Computer Graphics Forum*, pages 273–287. Wiley Online Library, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[3] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 7

[4] Enric Corona, Albert Pumarola, Guillem Alenya, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11875–11885, 2021. 2

[5] Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 8

[6] Artur Grigorev, Giorgio Becherini, Michael Black, Otmar Hilliges, and Bernhard Thomaszewski. Contourcraft: Learning to resolve intersections in neural multi-garment simulations. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 3, 5, 6, 8, 9

[7] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[8] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 18–35. Springer, 2020. 2

[9] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[10] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. *Proceedings of the fourth Eurographics symposium on Geometry processing*, 7:61–70, 2006. 4

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 2

[12] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4): 1–13, 2017. 7

[13] Maria Korosteleva and Sung-Hee Lee. Neuraltailor: Reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Trans. Graph.*, 41(4), 2022. 2

[14] Bruno Lévy and Nicolas Bonneel. Variational anisotropic surface meshing with voronoi parallel linear enumeration. In *Proceedings of the 21st international meshing roundtable*, pages 349–366. Springer, 2013. 4

[15] Yifei Li, Hsiao-yu Chen, Egor Larionov, Nikolaos Sarafianos, Wojciech Matusik, and Tuur Stuyck. Diffavatar: Simulation-ready garment optimization with differentiable simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4368–4378, 2024. 2

[16] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19711–19722, 2024. 3, 5, 8

[17] Siyou Lin, Boyao Zhou, Zerong Zheng, Hongwen Zhang, and Yebin Liu. Leveraging intrinsic properties for non-rigid garment alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14485–14496, 2023. 7

[18] Siyou Lin, Zhe Li, Zhaoqi Su, Zerong Zheng, Hongwen Zhang, and Yebin Liu. Layga: Layered gaussian avatars for animatable clothing transfer. In *SIGGRAPH Conference Papers*, 2024. 3

[19] Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2023. 2

[20] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 3

[21] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 7, 8

[22] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[23] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 3, 8

[24] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (ToG)*, 36(4):1–15, 2017. 2

[25] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20299–20309, 2024. 3, 4

[26] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting, 2024. 3

[27] Lingteng Qiu, Guanying Chen, Xu Mutian Zhou, Jiapeng, Junle Wang, and Xiaoguang Han. Rec-mv: Reconstructing 3d dynamic cloth from monocular videos. In *CVPR*, 2023. 2

[28] Igor Santesteban, Nils Thuerey, Miguel A Otaduy, and Dan Casas. 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*, pages 11763–11773, 2021. 8

[29] Igor Santesteban, Miguel A. Otaduy, and Dan Casas. Snug: Self-supervised neural dynamic garments, 2022. 5

[30] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[31] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1606–1616, 2024. 3

[32] PP Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020. 2

[33] Zhaoqi Su, Liangxiao Hu, Siyou Lin, Hongwen Zhang, Shengping Zhang, Justus Thies, and Yebin Liu. Caphy: Capturing physical properties for animatable human avatars. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[34] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 1–18. Springer, 2020. 2

[35] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 2

[36] Wenbo Wang, Hsuan-I Ho, Chen Guo, Boxiang Rong, Artur Grigorev, Jie Song, Juan Jose Zarate, and Otmar Hilliges. 4d-dress: A 4d dataset of real-world human clothing with semantic annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 550–560, 2024. 3, 6, 7

[37] Donglai Xiang, Fabian Prada, Timur Bagautdinov, Weipeng Xu, Yuan Dong, He Wen, Jessica Hodgins, and Chenglei Wu. Modeling clothing as a separate layer for an animatable human avatar. *ACM Transactions on Graphics (TOG)*, 40(6): 1–15, 2021. 2

[38] Donglai Xiang, Timur Bagautdinov, Tuur Stuyck, Fabian Prada, Javier Romero, Weipeng Xu, Shunsuke Saito, Jingfan Guo, Breannan Smith, Takaaki Shiratori, et al. Dressing avatars: Deep photorealistic appearance for physically simulated clothing. *ACM Transactions on Graphics (TOG)*, 41(6): 1–15, 2022. 2

[39] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 3

[40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[41] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, et al. Physavatar: Learning the physics of dressed 3d avatars from visual observations. *arXiv preprint arXiv:2404.04421*, 2024. 3

[42] Wang Zhou. Image quality assessment: from error measurement to structural similarity. *IEEE transactions on image processing*, 13:600–613, 2004. 7

[43] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 512–530. Springer, 2020. 2

[44] Heming Zhu, Lingteng Qiu, Yuda Qiu, and Xiaoguang Han. Registering explicit to implicit: Towards high-fidelity garment mesh reconstruction from single images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3845–3854, 2022. 2

[45] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. *arXiv preprint arXiv:2311.08581*, 2023. 3