Drivable 3D Gaussian Avatars

Wojciech Zielonka[1,2,3*], Timur Bagautdinov[3], Shunsuke Saito[3], Michael Zollhöfer[3], Justus Thies[1,2], Javier Romero[3]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Technical University of Darmstadt [3]Codec Avatars Lab, Meta
https://zielon.github.io/d3ga/

Figure 1. Given a multi-view video input, D3GA is trained to create light, drivable, photorealistic 3D human avatars. These avatars are constructed as a composition of 3D Gaussians encapsulated within tetrahedral cages. The Gaussians undergo transformation and stretching influenced by these cages, are colored using an MLP, and are rasterized into splats. By representing the drivable human as a collection of 3D Gaussian layers, we gain the ability to decompose and manipulate the avatar as needed.

Abstract

We present Drivable 3D Gaussian Avatars (D3GA), a multi-layered 3D controllable model for human bodies that utilizes 3D Gaussian primitives embedded into tetrahedral cages. The advantage of using cages compared to commonly employed linear blend skinning (LBS) is that primitives like 3D Gaussians are naturally re-oriented and their kernels are stretched via the deformation gradients of the encapsulating tetrahedron. Additional offsets are modeled for the tetrahedron vertices, effectively decoupling the low-dimensional driving poses from the extensive set of primitives to be rendered. This separation is achieved through the localized influence of each tetrahedron on 3D Gaussians, resulting in improved optimization. Using the cage-based deformation model, we introduce a compositional pipeline that decomposes an avatar into layers, such as garments, hands, or faces, improving the modeling of phenomena like garment sliding. These parts can be conditioned on different driving signals, such as keypoints for facial expressions or joint-angle vectors for garments and the body. Our experiments on two multi-view datasets with varied body shapes, clothes, and motions show higher-quality results. They surpass PSNR and SSIM metrics of other SOTA methods using the same data while offering greater flexibility and compactness.

1. Introduction

Developing drivable, photorealistic human avatars is crucial for better long-distance telecommunication that provides an immersive experience to the users. The motion and deformations across various segments of a complex avatar's body are influenced by distinct signals, such as facial expressions and body movements. This complexity poses challenges for accurate modeling using a single layer. Multi-layered avatars become essential to represent these different regions, ensuring proper motion and visual fidelity. Similarly, garments present challenges such as sliding, necessitating separate modeling of each clothing piece.

Mixture of Volumetric Primitives (MVP) [30] started a

*Work done while Wojciech Zielonka was an intern at the Codec Avatars Lab in Pittsburgh, PA, USA

successful line of hybrid implementations, where volumetric primitives are embedded on the surface of the tracked mesh. This representation, despite excellent results, struggles when the provided mesh is not precise or lacks details, ultimately producing artifacts and misaligning the primitives. Similar CNN-based architectures [1, 27, 29, 30, 52], do not allow for easy garment decomposition and assume a fixed amount of 3D primitives since the CNN size has to be set for the training. Furthermore, numerous methods [1, 24, 30, 57] lack the capability of layered conditioning specific to different body parts. For example, they may not support using keypoints for the face or motion vectors for clothing like t-shirts. This is an important aspect of a holistic system that, ultimately, needs to capture speech, face, gestures, and garment motion. State-of-the-art drivable avatars [52, 68] require dense input signals like RGB-D images or even multi-view camera setups at test time, which might not be suitable for low-bandwidth connections in telepresence applications. Finally, drivable NeRFs and 3DGS avatars typically rely on LBS to transform samples between canonical and observation spaces. However, LBS is limited by the low degree of freedom of the model, whereas cages can handle more complex non-linear motion and offer additional physical properties (e.g., stretching).

We designed our method to use a minimal set of inputs and still be competitive with the ones that require more information to train an avatar. D3GA models digital humans using volumetric primitives represented as 3D Gaussians embedded into a tetrahedral cage which is naturally described by phenomenons like stretching, rotation, and scaling. Accordingly, instead of LBS, our method builds on a classic deformation model for transforming volumes [40]. Specifically, by recasting cages from the canonical space into a deformed one, the 3D Gaussian covariance matrices undergo the encapsulating tetrahedral deformation transformation. Recent advancements in incorporating physics into Gaussians [8, 70] show further promise in the context of cage usage for garment modeling by capitalizing on [4, 35]. Also, cages decouple the representation resolution (related to the amount of Gaussians) from the degrees of freedom present in the model ultimately allowing an effective regularization of the deformations in contrast to LBS which depends on the global bone transformations only. In addition, we employ a compositional structure based on separate body, face, and garment cages, allowing us to model those parts independently, including localized conditioning based on different driving signals (e.g., keypoints).

We train person-specific models on nine high-quality multi-view sequences with a wide range of body shapes, motion, and clothing (not limited to tight-fitting), which later can be driven with new poses from any subject.

In summary, we present Drivable 3D Gaussian Avatars (D3GA) with the following contributions:

- A light, flexible, and composable model based on 3D Gaussian primitives driven by tetrahedral cage-based deformations which improve their body modeling properties.
- Localized motion conditioning which enables for instance facial expressions.

2. Related Work

D3GA reconstructs controllable digital full-body avatars using multi-view video and joint angle motion by combining 3D Gaussian Splatting (3DGS) [19] with cage-based deformations [12, 14, 17]. Current methods for controllable avatars rely on dynamic Neural Radiance Fields (NeRF) [38, 43, 44], point-based [34, 71, 77], or hybrid representations [1, 6, 30, 79], which are either slow to render or fail to correctly disentangle garments from the body, leading to poor generalization to new poses. Recently, incorporating 3DGS into dynamic scenarios has opened new research avenues [27, 49, 69, 72, 76]. For a thorough overview, we refer readers to state-of-the-art reports on digital avatars and neural rendering [60, 61, 82].

Dynamic Neural Radiance Fields NeRF [39] is a popular appearance model for human avatars, representing scenes volumetrically with density and color information using an MLP. Images are rendered via ray casting and volumetric integration of sample points [18]. Many methods have successfully applied NeRF to dynamic scenes [9, 26, 43, 44, 47, 65, 71, 79], achieving high-quality results. However, most methods treat avatars as a single layer [24, 38, 45, 55–57, 78], which complicates modeling phenomena like sliding or loose garments. Methods like [6, 7] address this using a hybrid representation, combining explicit geometry from SMPL[31] with implicit dynamic NeRF. Despite impressive garment reconstruction, these methods struggle with novel pose prediction. TECA [74] extends SCARF to a generative framework, enabling prompt-based generation of NeRF-based accessories and hairstyles.

Point-based Rendering Before 3DGS, many methods used point-based rendering [34, 57, 77] or sphere splatting [23], with optimizable positions and sizes. NPC by Su et al. [57] defines a point-based body model for avatar representation, but requires lengthy nearest neighbor searches during training (12 hours vs. 30 minutes for our model), making it impractical for dense multi-view datasets. Ma et al. [34] represent garments as a pose-dependent function mapping SMPL points [31] to the clothing space. This is improved in [48] with a neural deformation field, but both models only address geometry, not appearance. Zheng et al. [77] represent the upper part of an avatar as a point cloud, grown during optimization and rasterized using a differentiable renderer [63]. While achieving photorealistic local results, the avatars suffer from artifacts like holes.

Cage-based Deformations Cages[40] are commonly used for geometry modeling and animation, serving as sparse proxies to control all interior points, enabling efficient deformation by manipulating only cage nodes. Yifan et al. [64] introduced neural cages for detail-preserving shape deformation, where a neural network rigs the source object into the target via a proxy. Garbin et al. [10] extended dynamic NeRF with tetrahedron cages to unposed ray samples based on tetrahedron intersections. This method is real-time, high-quality, and controllable, but limited to objects with local deformations like heads, and not suitable for highly articulate objects like full-body avatars. Peng et al. used a cage to deform a radiance field in CageNeRF [46]. While their low-resolution cages can be applied to full-body avatars, they fail to model detailed features like faces or complex deformations.

Time-conditioned Methods Playback methods [2, 5, 13, 25, 66, 73] represent a scene as a time-conditioned function that cannot be arbitrarily controlled, allowing only for a novel viewpoint synthesis while traversing the time axis. Yang et al. [73] extended the representation of 3DGS [19] into 4DGS, effectively incorporating time into the primitive representation. Wu et al. [66] combine Gaussians with 4D neural voxels, inspired by HexPlane [2], which achieves real-time rendering and novel-view synthesis. However, these solutions fall into a different class of algorithms compared to pose-conditioned drivable avatars, which is our goal.

Dynamic Gaussian Splatting D3GA is based on 3D Gaussian Splatting (3DGS) [19], a recent alternative to NeRF for modeling neural scenes. Due to its real-time capabilities and high-quality results, 3DGS has inspired numerous follow-up papers [8, 15, 33, 49, 69, 70, 72, 76, 80, 81] in areas such as physics simulation, hair modeling, head avatars, and fluid dynamics. Several works [27, 41, 53] recently introduced convolutional networks to regress Gaussian maps. Despite achieving high-quality results, fixed convolutional architectures do not allow for local conditioning or adjusting the number of Gaussians during training. These methods also use up to 23 times more parameters, causing the model size to reach almost 1 GiB. In contrast, our pipeline remains lightweight and flexible, offering garment decomposition and localized conditioning. Finally, using CNNs can slow down the pipeline to around 10 FPS [27], whereas our method remains real-time.

3. Method

D3GA is built on 3DGS extended by a neural representation and tetrahedral cages to model the color and geometry of each dynamic part of the avatar, respectively. In the following, we introduce the formulation of 3D Gaussian Splatting and give a detailed description of our method.

3.1. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [19] is designed for real-time novel view synthesis in multi-view static scenes. Their rendering primitives are scaled 3D Gaussians [22, 63] with a 3D covariance matrix $\mathbf{\Sigma}$ and mean μ:

$$G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}. \tag{1}$$

To splat the Gaussians, Zwicker et al. [83] define the projection of 3D Gaussians onto the image plane as:

$$\mathbf{\Sigma}' = \mathbf{A}\mathbf{W}\mathbf{\Sigma}\mathbf{W}^T\mathbf{A}^T, \tag{2}$$

where $\mathbf{\Sigma}'$ is a covariance matrix in 2D space, $\mathbf{W}$ is the view transformation, and $\mathbf{A}$ is the Jacobian of the affine approximation of the projective transformation. During optimization, enforcing the positive semi-definiteness of the covariance matrix $\mathbf{\Sigma}$ is challenging. To avoid this, Kerbl et al. [19] use an equivalent formulation of a 3D Gaussian as a 3D ellipsoid parameterized with a scale $\mathbf{S}$ and rotation $\mathbf{R}$:

$$\mathbf{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T. \tag{3}$$

3DGS uses spherical harmonics [51] to model the view-dependent color of each Gaussian. In practice, appearance is modeled with an optimizable 48 elements vector representing four bands of spherical harmonics.

3.2. Body Cage Creation

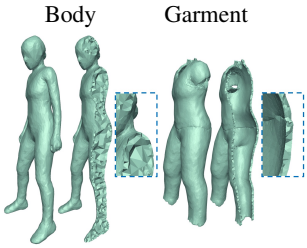


Figure 3. D3GA uses a tetrahedral mesh for deformation modeling.

To deform 3D Gaussians, we utilize tetrahedron cage-based deformations as a coarse proxy for the body, face, and individual garments. Unlike a triangle, which is two-dimensional, a tetrahedron is a polyhedron with four triangular faces (A, B, C, D), providing a three-dimensional structure. The volume of a tetrahedron can be calculated using the scalar triple product of vectors, which enables precise control and deformation of the enclosed 3D Gaussians. The volume V is given by:

$$V = \frac{1}{6}\left|\mathbf{AB} \cdot (\mathbf{AC} \times \mathbf{AD})\right| \tag{4}$$

where **AB, AC, AD** are edges of tetraherdon. This property allows us to compute the deformation gradient similarly to Sumner et al. [58] and transfer it to the Gaussian covariance matrix (Equation 7), see Supp. Mat, for more details.

To create a cage per garment, we segment all images of a single time instance using an EfficientNet [59] backbone

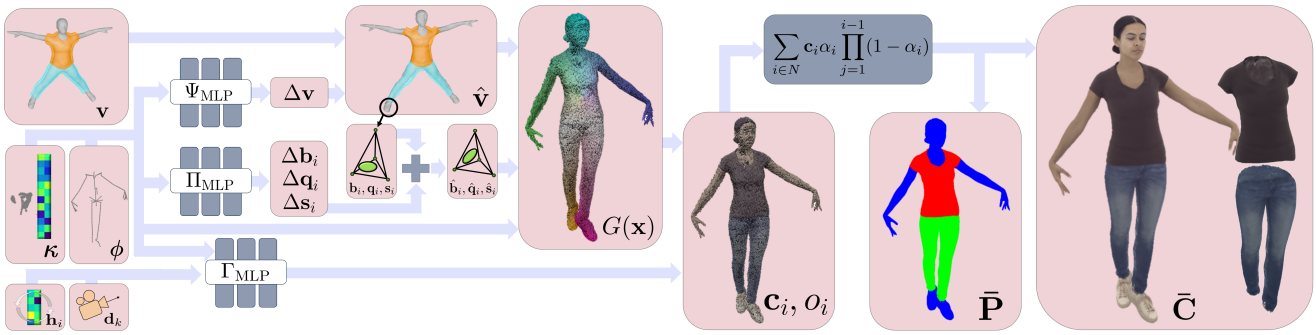


Figure 2. **Overview**. D3GA uses 3D pose $\boldsymbol{\phi}$, face embedding $\boldsymbol{\kappa}$, viewpoint $\mathbf{d}_k$ and canonical cage $\mathbf{v}$ (as well as auto-decoded color features $\mathbf{h}_i$) to generate the final render $\bar{\mathbf{C}}$ and auxiliary segmentation render $\bar{\mathbf{P}}$. The inputs in the left are processed through three networks ($\boldsymbol{\Psi}_{\text{MLP}}$, $\boldsymbol{\Pi}_{\text{MLP}}$, $\boldsymbol{\Gamma}_{\text{MLP}}$) per avatar part to generate cage displacements $\boldsymbol{\Delta}\mathbf{v}$, Gaussians deformations $\mathbf{b}_i$, $\mathbf{q}_i$, $\mathbf{s}_i$ and color/oppacity $\mathbf{c}_i$, o_i respectively. After cage deformations transform canonical Gaussians, they are rasterized into the final images according to Eq. 10.

with PointRend [21] refinement, trained on a corpus of similar multi-view captures. The per-image 2D segmentation masks are projected onto a body mesh $\hat{\mathbf{M}}$ to obtain per-triangle labels (body, upper, lower). To get the mesh $\hat{\mathbf{M}}$, we fit a low-resolution LBS model to a single 3D scan of the subject and then fit such model to the segmented frame by minimizing the distance to the 3D keypoints, extracted with an EfficientNet trained on similar captures. We transform the body mesh into canonical space with LBS and divide it into body part templates $\mathbf{M}_k$. The garment meshes are additionally inflated by 1-3 cm along the vertex normals. Afterward, we run a voxelization of the meshes and subsequently extract the mesh using the marching cubes algorithm [32]. After that, we use TetGen [54] to turn the unposed meshes $\mathbf{M}_k$ into tetrahedral meshes $\mathbf{T}_k$. Consequently, cages for garments are hollow, containing only their outer layer, while the body cage is solid (Figure 3). The face cage is composed of the body tetrahedra which contains triangles defined as the face on the LBS template. The cage nodes are deformed according to LBS weights transferred from the closest vertex in $\mathbf{M}_k$.

3.3. Cage Deformation Transfer

While classic cage methods typically deform the volume according to complex weight definitions [14, 16, 17], using linear weights works well in practice when cage cells are small, making it easier to integrate into an end-to-end training system. Specifically, we define $\mathbf{v}_{ij}$ as the vertices of tetrahedron i in canonical space, any point $\mathbf{x}$ inside this tetrahedron can be defined by its barycentric coordinates b_j:

$$\mathbf{x} = \sum_{j=1}^{4} b_j \mathbf{v}_{ij}. \tag{5}$$

Each Gaussian 3D mean $\boldsymbol{\mu} = \mathbf{x}$ is obtained as a linear combination of learnable barycentric coordinates b_j and tetrahedron vertices $\mathbf{v}_{ij}$. When the tetrahedra are transformed to posed space according to $\hat{\mathbf{v}}_{ij} = \text{LBS}(\mathbf{v}_{ij}, \boldsymbol{\phi}, \mathbf{w}_{ij})$, where $\boldsymbol{\phi}$ is the pose and $\mathbf{w}_{ij}$ are the blendweights, the same linear relation holds $\hat{\mathbf{x}} = \sum_{j=1}^{4} b_j \hat{\mathbf{v}}_{ij}$. To leverage the cage volume properties (rotation, sheer, and scaling), we use the deformation gradient [58]:

$$\mathbf{J}_i \mathbf{E}_i = \hat{\mathbf{E}}_i, \tag{6}$$

$$\mathbf{J}_i = \hat{\mathbf{E}}_i \mathbf{E}_i^{-1}, \tag{7}$$

where $\hat{\mathbf{E}}_i \in \mathbb{R}^{3\times3}$ and $\mathbf{E}_i \in \mathbb{R}^{3\times3}$ contain three edges from tetrahedron i defined in deformed and canonical spaces, respectively. The gradient $\mathbf{J}_i$ is used to transform the kernel of each Gaussian i (Eq 8). See Supp. mat for more details.

3.4. Drivable Gaussian Avatars

We initialize a fixed number of Gaussians, whose 3D means $\boldsymbol{\mu}$ are sampled on the surface of $\hat{\mathbf{M}}$. However, we are not limited to the fixed amount of Gaussians allowing for cloning or densification if needed. The rotation of each Gaussian is initialized so that the first two axes are aligned with the triangle surface and the third one with the normal: this is a good approximation for a smooth surface. The scale is initialized uniformly across a heuristic range depending on inter-point distances as in [19]. We assign each sampled position $\mathbf{x}$ to the intersecting tetrahedron and compute its barycentric coordinates $\mathbf{b} \in \mathbb{R}^4$. To deform the tetrahedron volume, we incorporate the deformation gradient $\mathbf{J}$ defined in Eq. 7 into the Gaussian covariance matrix from Eq. 3.

This is an important step as the deformation gradient $\mathbf{J}$ encapsulates many phenomena that we want to model, for instance, rotation, stretching, and sheering. To correctly transfer the deformation to 3D Gaussian primitives, we apply it to the covariance matrix $\boldsymbol{\Sigma}$, effectively modeling the 3DGS ellipsoids depending on the shape deformation from the canonical space into deformed one. Thus, the final covariance matrix passed to the rasterizer is denoted as:

$$\hat{\boldsymbol{\Sigma}} = \mathbf{J}_i \boldsymbol{\Sigma} \mathbf{J}_i^T, \tag{8}$$

where $\mathbf{J}_i$ is the deformation gradient of the tetrahedron containing the 3D mean of the Gaussian with covariance $\boldsymbol{\Sigma}$. This way, we transfer the deformation into the Gaussians, improving modeling phenomena like garment stretching.

Each part of the avatar (the garment, body, or face) is controlled by a separate GaussianNet $\mathbb{G}_{\text{Net}} = \{\Gamma_{\text{MLP}}, \Pi_{\text{MLP}}, \Psi_{\text{MLP}}\}$ which is defined as a set of small specialized multi-layer perceptrons (MLP) parametrized as:

$$\begin{aligned} \Psi_{\text{MLP}} &: \{\boldsymbol{\phi}, \text{enc}_{pos}(\mathbf{v})\} \rightarrow \Delta\mathbf{v}, \\ \Pi_{\text{MLP}} &: \{\boldsymbol{\phi}, \mathbf{b}_i, \mathbf{q}_i, \mathbf{s}_i\} \rightarrow \{\Delta\mathbf{b}_i, \Delta\mathbf{s}_i, \Delta\mathbf{q}_i\}, \\ \Gamma_{\text{MLP}} &: \{\boldsymbol{\phi}, \text{enc}_{view}(\mathbf{d}_k), \mathbf{h}_i, \mathbf{f}_j\} \rightarrow \{\mathbf{c}_i, o_i\}. \end{aligned} \tag{9}$$

All the networks take joint angles $\boldsymbol{\phi}$ (or face encodings $\boldsymbol{\kappa}$ for the face networks) as inputs, in addition to network-specific conditioning. The cage node correction network Ψ_{MLP} takes positional encodings [39] for all canonical vertices to transform them into offsets of the cage node positions similar to SMPL [31] pose-correctives. To adapt our representation further to the pose, the Gaussian correction network Π_{MLP} takes the canonical Gaussian parameters (barycentric coordinates $\mathbf{b}_i \in \mathbb{R}^4$, rotation $\mathbf{q}_i \in \mathbb{R}^4$ and scale $\mathbf{s}_i \in \mathbb{R}^3$) to predict corrections of those same parameters. These two networks are necessary to capture high-frequency details outside the parametric transformation.

The shading network Γ_{MLP} transforms encoded view direction and initial color into final color and opacity, $\mathbf{c}_i, o_i$. Unlike 3DGS, we use a pose-dependent color representation to model self-shadows and wrinkles in garments. The view angle is projected onto the first four spherical harmonics bands $\text{enc}_{pos}(\cdot)$, while the initial color is an auto-decoded feature vector $\mathbf{h}_i$ [42]. Additionally, the face region utilizes face embeddings $\boldsymbol{\kappa}$ as input instead of pose $\boldsymbol{\phi}$. This adaptability stems from our model's composability and holds the potential for extension to other regions, such as hair, shoes, or hands. A small auxiliary MLP regresses $\boldsymbol{\kappa}$ based on 150 3D keypoints $\mathbf{k}$ normalized by their training mean and standard deviations. This effectively enables us to model facial expressions.

Finally, we also add an embedding vector with the time frame of the current sample [36]. This allows D3GA to explain properties that cannot be modeled (e.g., cloth dynamics) from our input, effectively avoiding excessive blur due to averaging residuals. During testing, the average training embedding is used.

3.5. Training Objectives

As in 3DGS [19], we define the color $\bar{\mathbf{C}}$ of pixel (u, v):

$$\bar{\mathbf{C}}_{u,v} = \sum_{i \in \mathcal{N}} \mathbf{c_i} \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_i), \tag{10}$$

where $\mathbf{c_i}$ is the color predicted by Γ_{MLP}, which replaces the spherical harmonics in 3DGS. α_i is computed as the product of the Gaussian density in Eq. 1 with covariance matrix $\boldsymbol{\Sigma}'$ from Eq. 2 and the learned per-point opacity o_i predicted by Γ_{MLP}. The sum is computed over set $\mathcal{N}$, the Gaussians with spatial support on (u, v). The primary loss in D3GA is a weighted sum of three different color losses applied to the estimated image $\bar{\mathbf{C}}$ and the ground truth RGB image $\mathbf{C}$:

$$\mathcal{L}_{Color} = (1 - \omega)\mathcal{L}_1 + \omega\mathcal{L}_{\text{D-SSIM}} + \zeta\mathcal{L}_{\text{VGG}}, \tag{11}$$

where $\omega = 0.2$, $\zeta = 0.005$ (after 400k iterations steps and zero otherwise), $\mathcal{L}_{\text{D-SSIM}}$ is a structural dissimilarity loss, and $\mathcal{L}_{\text{VGG}}$ is the perceptual VGG loss.

To encourage correct garment separation, we introduce a garment loss. Since each Gaussian i is statically assigned to a part, we define $\mathbf{p_i}$ as a constant-per-part color and consequently render $\bar{\mathbf{P}}$ by replacing $\mathbf{c_i}$ by $\mathbf{p_i}$ in Eq. 10. Then, we compute the $\mathcal{L}_1$ norm between predicted parts $\bar{\mathbf{P}}$ and ground truth segmentations $\mathbf{P}$, $\mathcal{L}_{Garment} = \mathcal{L}_1(\bar{\mathbf{P}}, \mathbf{P})$. Moreover, we are using the Neo-Hookean loss based on Macklin et al. [35] to enforce the regularization of the predicted tetrahedra for the regions with low supervision signal:

$$\mathcal{L}_{Neo} = \frac{1}{N} \sum_{i=0}^{N} \frac{\lambda}{2} \left(\det(\mathbf{J}_i) - 1\right)^2 + \frac{\mu}{2} \left(\text{tr}(\mathbf{J}_i^T \mathbf{J}_i) - 3\right), \tag{12}$$

where $\mathbf{J}_i$ denotes the deformation gradient between a canonical and a deformed tetrahedron (Eq. 7), N is the total number of tetrahedrons, and λ and μ are the Lamé parameters [35]. The overall loss is defined as:

$$\mathcal{L} = \nu\mathcal{L}_{Color} + \nu\mathcal{L}_{Garment} + \tau\mathcal{L}_{Neo}, \tag{13}$$

where $\nu = 10$ and $\tau = 0.005$ balance the different losses.

We implemented D3GA based on the differentiable 3DGS renderer [19]. The networks Π_{MLP},Ψ_{MLP}, Γ_{MLP} have three hidden layers with 128 neurons and ReLU activation functions. In our experiments, we train the networks for 700k (Ours) and 400k (ActorsHQ) steps with a multi-step scheduler with a decay rate of 0.33, a batch size of one, and using the Adam optimizer [20] with a learning rate set to $5e-4$. We ran all experiments on a single Nvidia V100 GPU with 1024×667 images. When ground truth poses are not available, as in the case of ActorsHQ [13], we additionally refine poses regressed from keypoints during avatar training and during the test time, and optionally projected them onto PCA basis computed from the training set.

4. Dataset

Our dataset comprises nine subjects performing various motions, observed by 200 cameras. We use 12,000 frames for training (at 10 FPS) and 1,500 for testing (at 30 FPS). Images were captured at a resolution of 4096×2668 in a multi-view studio with synchronized cameras and downsampled to 1024×667 to reduce computational cost. We utilize 2D segmentation masks, RGB images, keypoints, and 3D joint angles for training, as well as a single registered mesh to create our template $\hat{\mathbf{M}}$. Of the nine subjects, data for four

Figure 4. Qualitative comparisons show that D3GA models facial expressions and garments better than other SOTA approaches. Especially regions with loose garments like skirts or sweatpants.

is publicly available through the Goliath-4 dataset release [37].

5. Results

We evaluate and benchmark our method w.r.t. five state-of-the-art multiview-based solutions [1, 11, 27, 30, 50]. We compare D3GA to the mesh-based full-body avatar methods BodyDecoder (**BD**) [1] and MVP-based avatars [30, 52] evaluated on our dataset.

Additionally, we evaluated D3GA on the ActorsHQ dataset [13] using a significantly smaller number of cameras (40). We compare to SOTA pose-conditioned 3DGS avatar methods, including Animatable Gaussians (**AG**) [27], 3DGS-Avatar [50], and Gaussian Avatar (**GA**) [11] which were trained on the same multiview data.

Please note that our method, along with 3DGS-Avatar and GA, represents a lightweight class of MLP-based algorithms, utilizing up to 10 million parameters. In contrast, the CNN-based MVP, BD, and AG [27] which in this case uses approximately 23 times more parameters (230 million).

5.1. Image Quality Evaluation

Our model is evaluated using SSIM, PSNR, and the perceptual metric LPIPS [75], with random color backgrounds. For the ActorsHQ evaluation, we utilized SMPL-X fittings obtained through OpenPose [3] and scan-to-mesh optimization. Table 1 shows that our method achieves the best PSNR and SSIM on our dataset compared to MVP [30] and BD [1]. Furthermore, on the ActorsHQ dataset, D3GA outperforms other Gaussian Avatar methods in terms of PSNR and SSIM. However, similar to previous evaluations, our method lacks sharpness due to its much smaller size compared to the CNN-based architecture of AG [27]. Moreover, our approach allows us to decompose avatars into drivable layers, unlike other volumetric methods. Each separate garment layer can be controlled solely by skeleton joint angles, without requiring specific garment registration modules as in [67].

Dataset	Method	PSNR ↑	LPIPS ↓	SSIM ↑
Ours	Ours	**30.634**	0.054	**0.964**
	MVP [30]	28.795	0.051	0.955
	BD [1]	29.918	**0.044**	0.959
ActorsHQ	Ours	**26.562**	0.065	**0.944**
	GA [11]	24.731	0.088	0.933
	3DGS-Avatar [50]	21.709	0.082	0.915
	AG [28]	26.454	**0.055**	0.937

Table 1. Our method scores the best in terms of PSNR and SSIM compared to BD [1] and MVP [30] on our dataset. D3GA is the best among MLP-based avatars, ranking only second in terms of sharpness compared to AG, which uses a CNN-based architecture.

Figure 5. ActorsHQ [13] comprises challenging garments that contain high-frequency patterns. Our method despite its small size can capture it and performs the best in terms of PSNR and SSIM, ranking second only in terms of sharpness to AG [27], which presents very sharp results due to the powerful StyleUNet [62].

5.2. Ablation Studies

Importance of cage deformations We replaced tetrahedrons with triangles to emphasize the crucial role of cage deformation gradients in transforming Gaussians. We modified Eq 5 such that 3D means are obtained through the barycentric coordinates of triangles $\mathbf{b} \in \mathbb{R}^3$ instead of tetrahedrons $\mathbf{b} \in \mathbb{R}^4$. The rest of the pipeline remains unchanged, with MLPs computing the same corrective terms as our cage-based model. Since triangles do not provide volume, we disabled the application of the cage deformation gradient $\mathbf{J}$, but the Gaussians are still modeled by the predicted residuals w.r.t. the canonical space. Figure 8 shows that the triangle-based approach does not stretch the primitives correctly, creating holes and artifacts which demonstrates the importance of using cages for deformation.

Garment loss The garment loss $\mathcal{L}_{Garment}$ (Fig. 7) serves two primary purposes: it improves garment separation and reduces erroneously translucid regions. We can observe qualitatively that regions between garments' boundaries without the regularizer are blurry and have erroneous opac-

Figure 6. D3GA enables motion transfer showing good generalizability while preserving each avatar's high-quality details.

Method	#parameters (M)	size (MiB)
Ours	9	45
GaussianAvatar [11]	7	59
3DGS-Avatar [50]	6	57
AG [28]	232	862

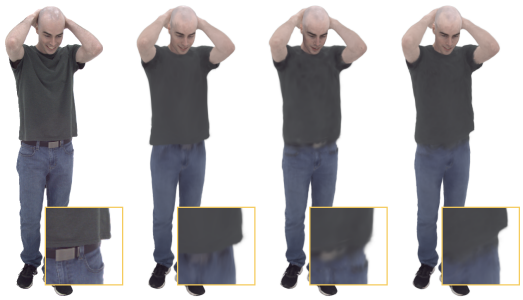
Table 2. Model compactness. D3GA offers the best tradeoff between quality and model size.

ity, see supp. mat. **Single layer avatar** D3GA supports a single-layer training for the garment and body, which struggles to model proper garment sliding. The results are presented in the last column of Fig. 7. It can be observed that the edges between the T-shirt and jeans are over-smoothed.

Size and compactness Our model offers an optimal balance between quality and model size, making it both compact and easily portable. This lightweight representation sets D3GA apart from much larger and more cumbersome models like AG [27]. As shown in Table 2, D3GA is similar in size to other methods, yet it delivers superior quality compared to models in the same category. This makes D3GA an attractive choice for telepresence applications, where both efficiency and performance are crucial.

6. Discussion

While D3GA shows better quality and competitive rendering speed w.r.t. the state of the art, there are still particular challenges. High-frequency patterns, like stripes, may result in blurry regions. One way of improving image quality would be using a variational autoencoder to regress Gaussian parameters per texel of a guide mesh similar to [27, 30].



Ground Truth Ours w/o $\mathcal{L}_{Garment}$ Single Layer

Figure 7. Ablation of D3GA: shape smoothness without $\mathcal{L}_{Garment}$, and sliding artifacts with a single layer representation.

Figure 8. Gaussian primitives embedded in triangles, compared to tetrahedrons, produce more artifacts, resulting in small holes and reduced sharpness that is reflected in the LIPIS score, which drops from 0.0648 to 0.0703.

Despite using the $\mathcal{L}_{Garment}$ loss, self-collisions for loose garments are still challenging, and the sparse controlling signal does not contain enough information about complex wrinkles or self-shadowing. A potential solution to solve self-penetration would be to incorporate explicit collision detection [4] for the tetrahedrons. An exciting follow-up work direction would be replacing the appearance model in D3GA with a relightable one. D3GA is currently limited to model photorealistic avatars for a few consenting subjects captured in a dense multi-view capture device. While this limits the potential misuse of the technology of driving somebody else's avatar without their consent, it needs to be addressed in future work. In conclusion, it's worth noting that the D3GA offers significant flexibility and can be customized for particular applications. For instance, one could employ additional Gaussians to capture high-frequency detail or opt to eliminate garment supervision, particularly if precise cage geometry decomposition isn't necessary.

7. Conclusion

We have proposed D3GA, a novel approach for reconstructing multi-layered animatable human avatars using tetrahedral cages embedded with 3D Gaussians. To transform the rendering primitives from canonical to deformed space, we directly apply the deformation gradient to the 3D Gaussian parametrization, enabling improved avatar modeling. Our method's compositional approach enables various forms of localized conditioning, such as using keypoints for facial expressions, and can be extended to other regions like hair, hands, or shoes. This capability is essential for creating holistic avatars driven by diverse input signals. We have demonstrated high-quality results that surpass state-of-the-art methods with similar model architectures, all while maintaining a lightweight, real-time, and compact approach.

References

[1] Timur M. Bagautdinov, Chenglei Wu, Tomas Simon, Fabián Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason M. Saragih. Driving-signal aware full-body avatars. *ACM Transactions on Graphics (TOG)*, 40:1 – 17, 2021. 2, 6

[2] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6

[4] Heng Chen, Elier Diaz, and Cem Yuksel. Shortest path to boundary for self-intersecting meshes. *ACM Transactions on Graphics (TOG)*, 42:1 – 15, 2023. 2, 8

[5] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 3

[6] Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J. Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. *SIGGRAPH Asia 2022 Conference Papers*, 2022. 2

[7] Yao Feng, Weiyang Liu, Timo Bolkart, Jinlong Yang, Marc Pollefeys, and Michael J. Black. Learning disentangled avatars with hybrid 3d representations. *arXiv*, 2023. 2

[8] Yutao Feng, Xiang Feng, Yintong Shang, Ying Jiang, Chang Yu, Zeshun Zong, Tianjia Shao, Hongzhi Wu, Kun Zhou, Chenfanfu Jiang, and Yin Yang. Gaussian splashing: Dynamic fluid synthesis with gaussian splatting. *ArXiv*, abs/2401.15318, 2024. 2, 3

[9] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8645–8654, 2020. 2

[10] Stephan J. Garbin, Marek Kowalski, Virginia Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, Jingjing Shen, Matthew Johnson, and Julien Valentin. Voltemorph: Real-time, controllable and generalisable animation of volumetric representations. *CoRR*, abs/2208.00949, 2022. 3

[11] 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. 6, 7, 8

[12] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Harry Shum. Subspace gradient domain mesh deformation. *ACM SIGGRAPH 2006 Papers*, 2006. 2

[13] Mustafa Isik, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Trans. Graph.*, 42(4):160:1–160:12, 2023. 3, 5, 6, 7

[14] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine-Hornung. Bounded biharmonic weights for real-time deformation. *ACM SIGGRAPH 2011 papers*, 2011. 2, 4

[15] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. *ArXiv*, abs/2401.16663, 2024. 3

[16] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26(3):71, 2007. 4

[17] Tao Ju, Scott Schaefer, and Joe D. Warren. Mean value coordinates for closed triangular meshes. *ACM SIGGRAPH 2005 Papers*, 2005. 2, 4

[18] James T. Kajiya. The rendering equation. *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986. 2

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 2, 3, 4, 5

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[21] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross B. Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9796–9805. Computer Vision Foundation / IEEE, 2020. 4

[22] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum*, 40, 2021. 3

[23] Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1440–1449, 2021. 2

[24] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. *ArXiv*, abs/2206.08929, 2022. 2

[25] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[26] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6494–6504, 2020. 2

[27] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian

maps for high-fidelity human avatar modeling. *ArXiv*, abs/2311.16096, 2023. 2, 3, 6, 7, 8

[28] 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 (CVPR)*, 2024. 6, 8

[29] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor. *ACM Transactions on Graphics (TOG)*, 40:1 – 16, 2021. 2

[30] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason M. Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40:1 – 13, 2021. 1, 2, 6, 8

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2015. 2, 5

[32] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pages 163–169, 1987. 4

[33] Haimin Luo, Ouyang Min, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. Gaussianhair: Hair modeling and rendering with light-aware gaussians. *ArXiv*, abs/2402.10483, 2024. 3

[34] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10954–10964, 2021. 2

[35] Miles Macklin and Matthias Müller. A constraint-based formulation of stable neo-hookean materials. *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2021. 2, 5

[36] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7206–7215, 2020. 5

[37] Julieta Martinez, Emily Kim, Javier Romero, Timur Bagautdinov, Shunsuke Saito, Shoou-I Yu, Stuart Anderson, Michael Zollhöfer, Te-Li Wang, Shaojie Bai, Chenghui Li, Shih-En Wei, Rohan Joshi, Wyatt Borsos, Tomas Simon, Jason Saragih, Paul Theodosis, Alexander Greene, Anjani Josyula, Silvio Mano Maeta, Andrew I. Jewett, Simon Venshtain, Christopher Heilman, Yueh-Tung Chen, Sidi Fu, Mohamed Ezzeldin A. Elshaer, Tingfang Du, Longhua Wu, Shen-Chi Chen, Kai Kang, Michael Wu, Youssef Emad, Steven Longay, Ashley Brewer, Hitesh Shah, James Booth, Taylor Koska, Kayla Haidle, Matt Andromalos, Joanna Hsu, Thomas Dauer, Peter Selednik, Tim Godisart, Scott Ardisson, Matthew Cipperly, Ben Humberston, Lon Farr, Bob Hansen, Peihong Guo, Dave Braun, Steven Krenn, He Wen, Lucas Evans, Natalia Fadeeva, Matthew Stewart, Gabriel Schwartz, Divam Gupta, Gyeongsik Moon, Kaiwen Guo, Yuan Dong, Yichen Xu, Takaaki Shiratori, Fabian Prada, Bernardo R. Pires, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, and Yaser Sheikh. Codec Avatar Studio: Paired Human Captures for Complete, Driveable, and Generalizable Avatars. *NeurIPS Track on Datasets and Benchmarks*, 2024. 6

[38] Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. Keypointnerf: Generalizing image-based volumetric avatars using relative spatial encoding of keypoints. *ArXiv*, abs/2205.04992, 2022. 2

[39] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf. *Communications of the ACM*, 65:99 – 106, 2020. 2, 5

[40] Jesús R Nieto and Antonio Susín. Cage based deformations: a survey. In *Deformation Models: Tracking, Animation and Applications*, pages 75–99. Springer, 2012. 2, 3

[41] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. Ash: Animatable gaussian splats for efficient and photoreal human rendering. *ArXiv*, abs/2312.05941, 2023. 3

[42] 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. 5

[43] Keunhong Park, U. Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5845–5854, 2020. 2

[44] Keunhong Park, U. Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf. *ACM Transactions on Graphics (TOG)*, 40:1 – 12, 2021. 2

[45] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9050–9059, 2020. 2

[46] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. In *NeurIPS*, 2022. 3

[47] Malte Prinzler, Otmar Hilliges, and Justus Thies. Diner: Depth-aware image-based neural radiance fields. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12449–12459, 2022. 2

[48] Sergey Prokudin, Qianli Ma, Maxime Raafat, Julien Valentin, and Siyu Tang. Dynamic point fields. *arXiv preprint arXiv:2304.02626*, 2023. 2

[49] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians, 2023. 2, 3

[50] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. 2024. 6, 7, 8

[51] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001. 3

[52] Edoardo Remelli, Timur M. Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabián Prada, Jason M. Saragih, and Yaser Sheikh. Drivable volumetric avatars using texel-aligned features. *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. 2, 6

[53] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. *ArXiv*, abs/2312.03704, 2023. 3

[54] Hang Si. Tetgen: A quality tetrahedral mesh generator and a 3d delaunay triangulator (version 1.5 — user's manual). 2013. 4

[55] Shih-Yang Su, Frank Yu, Michael Zollhoefer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. In *Neural Information Processing Systems*, 2021. 2

[56] Shih-Yang Su, Timur M. Bagautdinov, and Helge Rhodin. Danbo: Disentangled articulated neural body representations via graph neural networks. In *European Conference on Computer Vision*, 2022.

[57] Shih-Yang Su, Timur M. Bagautdinov, and Helge Rhodin. Npc: Neural point characters from video. *ArXiv*, abs/2304.02013, 2023. 2

[58] Robert W. Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM SIGGRAPH 2004 Papers*, 2004. 3, 4

[59] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6105–6114. PMLR, 2019. 3

[60] Ayush Tewari, Otto Fried, Justus Thies, Vincent Sitzmann, S. Lombardi, Z. Xu, Tanaba Simon, Matthias Nießner, Edgar Tretschk, L. Liu, Ben Mildenhall, Pranatharthi Srinivasan, R. Pandey, Sergio Orts-Escolano, S. Fanello, M. Guang Guo, Gordon Wetzstein, J y Zhu, Christian Theobalt, Manju Agrawala, Donald B. Goldman, and Michael Zollhöfer. Advances in neural rendering. *Computer Graphics Forum*, 41, 2021. 2

[61] Ayush Kumar Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason M. Saragih, Matthias Nießner, Rohit Pandey, S. Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B. Goldman, and Michael Zollhofer. State of the art on neural rendering. *Computer Graphics Forum*, 39, 2020. 2

[62] Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. Styleavatar: Real-time photo-realistic portrait avatar from a single video. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 7

[63] Yifan Wang, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38:1 – 14, 2019. 2, 3

[64] Yifan Wang, Noam Aigerman, Vladimir G. Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 72–80. Computer Vision Foundation / IEEE, 2020. 3

[65] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16189–16199. IEEE, 2022. 2

[66] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

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

[68] Donglai Xiang, Fabián Prada, Zhe Cao, Kaiwen Guo, Chenglei Wu, Jessica K. Hodgins, and Timur M. Bagautdinov. Drivable avatar clothing: Faithful full-body telepresence with dynamic clothing driven by sparse rgb-d input. 2023. 2

[69] Jun Xiang, Xuan Gao, Yudong Guo, and Ju yong Zhang. Flashavatar: High-fidelity digital avatar rendering at 300fps. *ArXiv*, abs/2312.02214, 2023. 2, 3

[70] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics, 2023. 2, 3

[71] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5428–5438, 2022. 2

[72] Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. *ArXiv*, abs/2312.03029, 2023. 2, 3

[73] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2024. 3

[74] Hao Zhang, Yao Feng, Peter Kulits, Yandong Wen, Justus Thies, and Michael J. Black. Text-guided generation and editing of compositional 3d avatars. *ArXiv*, abs/2309.07125, 2023. 2

[75] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[76] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. *ArXiv*, abs/2312.02155, 2023. 2, 3

[77] Yufeng Zheng, Yifan Wang, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21057–21067, 2022. 2

[78] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15872–15882, 2022. 2

[79] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4574–4584, 2022. 2

[80] Wojciech Zielonka, Timo Bolkart, Thabo Beeler, and Justus Thies. Gaussian eigen models for human heads. *arXiv:2407.04545*, 2024. 3

[81] Wojciech Zielonka, Stephan J. Garbin, Alexandros Lattas, George Kopanas, Paulo Gotardo, Thabo Beeler, Justus Thies, and Timo Bolkart. Synthetic prior for few-shot drivable head avatar inversion. *arXiv:2501.06903*, 2025. 3

[82] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. *Computer Graphics Forum*, 37, 2018. 2

[83] Matthias Zwicker, Hans Rüdiger Pfister, Jeroen van Baar, and Markus H. Gross. Surface splatting. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001. 3