Human Gaussian Splatting: Real-time Rendering of Animatable Avatars

Arthur Moreau* Jifei Song* Helisa Dhamo Richard Shaw
Yiren Zhou Eduardo Pérez-Pellitero
Huawei Noah's Ark Lab

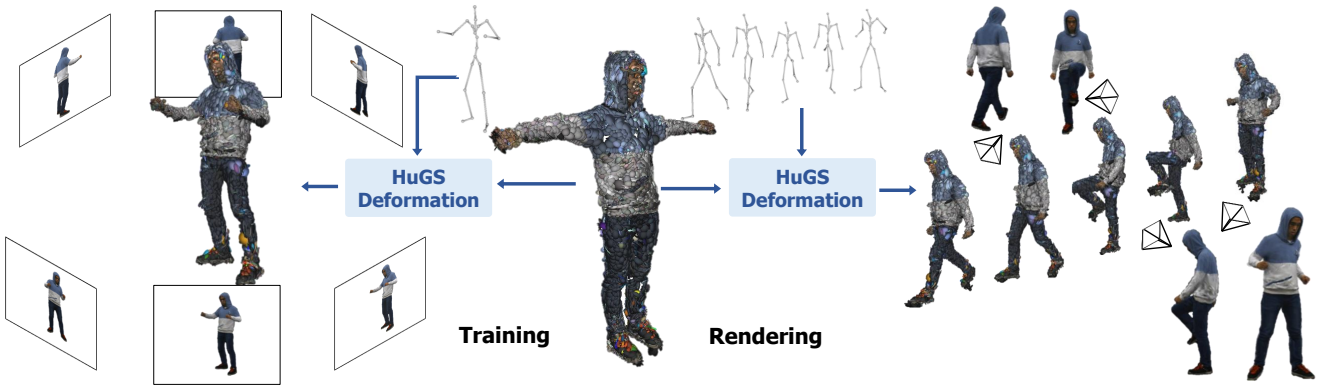

Figure 1. **Overview of HuGS.** Using multi-view video frames of a dynamic human, HuGS learns a photorealistic 3D human avatar represented by a 3D Gaussian Splatting model. Given an arbitrary human body pose, our model deforms the canonical 3D representation to the observation space, from which novel views can be rendered in real-time from any camera viewpoint.

Abstract

This work addresses the problem of real-time rendering of photorealistic human body avatars learned from multi-view videos. While the classical approaches to model and render virtual humans generally use a textured mesh, recent research has developed neural body representations that achieve impressive visual quality. However, these models are difficult to render in real-time and their quality degrades when the character is animated with body poses different than the training observations. We propose an animatable human model based on 3D Gaussian Splatting, that has recently emerged as a very efficient alternative to neural radiance fields. The body is represented by a set of gaussian primitives in a canonical space which is deformed with a coarse to fine approach that combines forward skinning and local non-rigid refinement. We describe how to learn our Human Gaussian Splatting (HuGS) model in an end-to-end fashion from multi-view observations, and evaluate it against the state-of-the-art approaches for novel pose synthesis of clothed body. Our method achieves 1.5 *dB PSNR improvement over the state-of-the-art on THuman4 dataset while being able to render in real-time (*≈ 80 *fps for* 512×512 *resolution).*

1. Introduction

Virtual human avatars are essential components of virtual reality and video games for applications such as content creation and immersive interaction between users and virtual worlds. The common procedure to create a highly realistic avatar of a person involves expensive sensors and tedious manual work. However, recent progress in 3D modeling and neural rendering enabled data-driven models that learn controllable human avatars from images.

Recent research in this area focuses on neural representations based on Neural Radiance Fields (NeRF) [36] or Signed Distance Fields (SDF) [40]. Thanks to their continuous design, they can represent highly detailed shape and textures of a clothed human body, and thus exhibit better quality than the commonly-used textured meshes. However, their deployment is difficult, first because they present long training and rendering times, but also because animating the character in a controllable way is challenging. As a result, these neural avatars shine on novel view synthesis of body poses observed during training, but struggle to generalize to novel body poses with similar quality.

While animating meshes is a well understood problem

*Authors contributed equally to this work.

that has been around for decades, *e.g*. using direct skinning algorithms [15, 20, 35], transferring these ideas for implicit neural representations is challenging. Some efforts have been made to learn skinning weights fields, but they often define a backward skinning methodology [23, 43, 60], which is pose-dependent and where solutions are slow and do not generalize well to unseen body poses.

In this work, we represent the human body with 3D Gaussian Splatting (3D-GS) [16]. This novel paradigm for view synthesis uses an explicit set of primitives shaped as 3D Gaussians to represent the radiance field of a scene. This enables fast tile-based rasterization, which is orders of magnitudes faster than the rendering speed of implicit methods based on ray marching. Leveraging its explicit and discrete nature, we explore its capability to be deformed with direct forward skinning, similar to a mesh. We observe that using Linear Blend Skinning (LBS) with skinning weights learned for each Gaussian provides an efficient and effective method that generalizes well to new body poses, but is not expressive enough to capture the local garment deformations of clothed avatars. We propose to refine the deformation of Gaussians with a shallow neural network that captures the local movements of the body surface.

The proposed algorithm, named **Human Gaussian Splatting (HuGS)**, paves the way for animatable human body models based on Gaussian Splatting. Our contributions can be summarized as follows:

1. We propose a first algorithm for novel pose synthesis of the human body based on 3D Gaussian Splatting.
2. We define a coarse-to-fine approach to animate the set of Gaussian primitives, based on forward skinning for skeleton-based movements and a pose-dependent MLP for local garments deformations.
3. We compare HuGS to neural-based human models on three public datasets and exhibit results on-par or better than state-of-the-art methods, while rendering one order of magnitude faster.

2. Related Work

Differentiable rendering of radiance fields Learning 3D representations from 2D images for novel view synthesis has been very active in the last few years [11, 37, 64] since the seminal work of Neural Radiance Fields [36]. While NeRF methods usually define emitted color and density of each point of a static scene, it has been adapted to model dynamic content by incorporating the time dimension in the representation [24, 41, 47]. These dynamic radiance fields can be used to train volumetric representation of humans in movement [10, 45] and to replay an existing video from a new camera viewpoint. NeRF and related approaches rely on ray marching and volumetric rendering [58], requiring to evaluate a neural network on many points along each camera ray. Making this evaluation more efficient can be tackled by storing information in an explicit way [2, 18, 25, 38, 52, 57, 65] but the ray marching design inherently limits the rendering time of these methods.

By contrast, 3D Gaussian Splatting [16] models the radiance field of a scene with explicit primitives shaped as 3D gaussians. The main advantage comes from the fast rendering step that avoids ray marching by sorting and splatting primitives w.r.t. the camera position, enabling real-time applications. Each primitive is defined by its position, covariance matrix, view-dependent color represented with spherical harmonics, and opacity. These parameters are optimized through gradient descent to reconstruct the observed images with high fidelity. Recent works have extended 3D-GS to dynamic scenes by learning time-dependent gaussian parameters [33, 53, 63, 66, 67]. Instead, our work focuses on a model tailored for a drivable human by learning gaussians deformations that depend on the body pose.

Animatable human body models We refer to an *animatable body model* as a 3D representation of a human character, which is defined in a canonical space (i.e. a reference body pose, often set as T-pose) and can be deformed to represent any body pose in their respective observation (or posed) space. Pioneer works in human body modelling are statistical mesh templates such as SMPL [32, 42]. Fitted to 3D scans of a large set of people, they provide a parametric representation of the human body shape. The deformation from canonical to observation space can be computed easily with linear blend skinning and thus these templates are often used as a building block of more complex methods. Avatars can be learned from different data modalities, such 3D scans [4, 51, 56], monocular videos [62, 69] or a single image [3, 28]. We present below methods that use multi-view RGB video capture.

Recent literature has explored the use of neural representations, either based on NeRF or SDF. These methods perform ray marching in the observation space but usually define a static radiance field in the canonical space. One approach is to establish backward correspondences (from observation to canonical) for each point. Neural Actor [29], Animatable NeRF [43] and InstantNVR [7] use pose-dependent networks to learn deformation or blend-weights fields, but they have been observed to generalize poorly to novel body poses. ARAH [60], TAVA [23] and PoseVocab [26] rather use a joint root-finding approach that generalizes better but is slow to compute.

Another line of work, closer to ours, circumvent the backward correspondence problem by applying forward skinning deformation on different kind of canonical *primitives*, resulting in a radiance field defined in the observation space. Neural Body [44] anchors latent codes to SMPL vertices which are then decoded to density and colors by a CNN. SLRF [72] and AvatarRex [73] use mul-

tiple local NeRFs centered on vertices whose origins are moved with LBS. DVA [50] computes deformation of articulated volumetric primitives [31]. Several methods use point-based primitives, for which forward skinning is well defined, either with volumetric rendering [55, 68] or rasterization [46, 71] approaches. Overall, forward skinning is more convenient to use than backward approaches. However, by defining the primitives on the SMPL template surface, such approaches often struggle to represent characters with loose clothing.

Finally, skeleton-based rigid deformations are not a sufficient driving signal to explain all the movement observed in the data [1]. Local garment wrinkles or muscles contractions are typical examples of non-rigid motion that need to be addressed. Modelling these phenomenons in a proper way would require physics-based modelling [56], but this remains difficult together with RGB supervision. Most learning-based methods rather use local refinement of the geometry with pose-dependent neural networks [23, 26, 72]. Some methods also add local shading components [1, 23] to approximate the ambient occlusion used in graphics pipelines.

3. Method

Our goal is to reconstruct human avatars from multi-view videos and render images of the reconstructed virtual character from arbitrary camera view and body poses, *i.e.*, novel pose synthesis. An overview of our method is presented in Figure 2. We train our model from a collection of multi-view videos depicting various body poses, captured by N calibrated cameras during T timesteps. We pre-compute or assume access to SMPL [32] or SMPL-X [42] parameters, *i.e.*, body shape $\boldsymbol{\beta}$ and body poses $\boldsymbol{\theta_t} = (\theta_1, \theta_j, \dots, \theta_J)$ expressed as the 3D rotation of each body joint j, at timestep t. We also use foreground segmentation masks.

3.1. Canonical representation

We represent the canonical human body as a set of volumetric primitives shaped as 3D Gaussians. Each Gaussian is parametrized by its own set of learnable parameters.

- a 3D canonical position $\mathbf{p}_c = (p_x, p_y, p_z)$,
- a 3D orientation, represented by a quaternion $\mathbf{q}_c$,
- a 3D scale $\mathbf{s} = (s_x, s_y, s_z)$,
- a color $\mathbf{c} = (c_r, c_g, c_b)$,
- an opacity scalar value o,
- a skinning weight vector $\mathbf{w} = (w_1, w_j, \dots, w_J)$ that regulates the influence of each body joint j on how the gaussian moves,
- a latent code $\mathbf{l}$ that encodes the non-rigid motion.

This builds up from the original 3D Gaussian splatting formulation [16], with the addition of the last 2 parameters that encode the pose-dependent movement of each primitive. We consider scale and opacity consistent across novel views and novel poses. For a given target body pose, we transform canonical position, orientation, and color to the posed space, as described in sections 3.2 and 3.3.

3.2. Deformation with forward skinning.

We use Linear Blend Skinning (LBS) to deform our model. We consider body joints in the canonical space imported from the SMPL template model. Given a body pose $\boldsymbol{\theta_t}$, we can compute the rigid transformations $\mathbf{M_j} \in \mathrm{SE}(3)$ for the j-th body joint using the kinematic tree. Then, each gaussian's skinning transformation $\mathbf{T_t}$ for pose $\boldsymbol{\theta_t}$ is defined by weight-averaging joint transformations according to skinning weights $\mathbf{w}$:

$$\mathbf{T_t} = \sum_{j=1}^{J} w_j \mathbf{M_j}. \tag{1}$$

The canonical Gaussian position is then transformed to the posed space using $\mathbf{T_t}$. We also rotate the gaussian using the rotation component $\mathbf{R_t}$ of $\mathbf{T_t} = [\mathbf{R_t}|\mathbf{t_t}]$:

$$\mathbf{p}_{lbs} = \mathbf{T_t}\mathbf{p}_c \qquad \mathbf{q_{lbs}} = \mathbf{R_t} \circ \mathbf{q}_c. \tag{2}$$

We apply directly forward skinning on the canonical primitives, similar to mesh deformation in common pipelines, and learn only the skinning weights attached to each Gaussian. In contrast, previous NeRF and SDF approaches, because they rely on ray marching on the posed space, need backward skinning formulations [23, 43, 60] which is notoriously more difficult and/or slower.

3.3. Local non-rigid refinement

LBS moves the Gaussians towards the target body pose and provides excellent generalization to novel poses, but only encodes the rigid deformations of the body joints. Because we want our method to be able to operate on clothed avatars, we also need to model local non-rigid deformations caused by garments. To this end, we compute per-gaussian residual outputs learned by a pose-dependent MLP that can translate, rotate, and change the lightness of the primitive.

Body pose encoding We want our model to learn how Gaussians move w.r.t. the body pose rather than w.r.t. time, but we also expect the network to learn *local* deformations. Thus, using the global pose vector $\boldsymbol{\theta_t}$ as input can represent too many information for local primitives whose deformation depends on a nearby joints orientations only. This can ultimately enable the network to learn spurious correlations and overfit [26]. Following SCANimate [51], we use an attention-weighting scheme which uses the skinning weights $\mathbf{w}$ and an explicit attention map W based on the kinematic tree to define the local pose vector $\boldsymbol{\theta}_t^l$:

$$\boldsymbol{\theta_t^l} = (W \cdot \mathbf{w}) \odot \boldsymbol{\theta_t}, \tag{3}$$

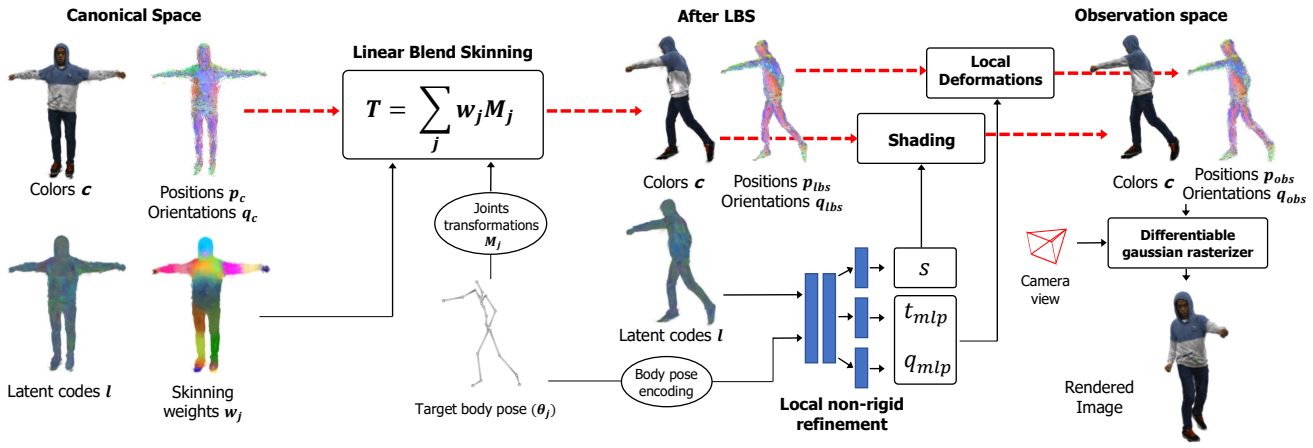

Figure 2. **HuGS method overview.** We represent the different attributes of Gaussians at each step of the deformation pipeline. Canonical positions and orientations are first deformed with LBS using the learned skinning weights. Then positions, orientations, and colors are refined by an MLP using the latent codes. Finally, Gaussians in the observation space are rendered through the target camera view.

where $\boldsymbol{\theta_t}$ is represented as a vector of quaternions and $\odot$ denotes element-wise multiplication.

Shading While our approach does not model view-dependent specularities because we consider the human body as Lambertian, we allow the color to change depending on the local body pose. This enables us to take into account self-occlusions and shadows caused by garment wrinkles. This shading approximates ambient occlusion in graphics pipelines [34]. Formally, the MLP outputs a scaling factor $s \in [0, 2]$ that multiplies the RGB color of each Gaussian, which is then clipped in $[0, 1]$.

MLP architecture Our neural network takes as input the local body pose vector $\boldsymbol{\theta_t}$ concatenated with the per-gaussian learnable latent features $\mathbf{l}$. This latent code identifies each primitive and encodes their local motion depending on the body pose which is then decoded by the MLP. We show in Section 4.4 that using this latent code leads to better performance than using the position. This information is processed by 2 hidden layers with 64 neurons and ReLU activations and then decoded by 3 separate heads with 2 layers that output respectively a translation vector $\mathbf{t}_{\text{mlp}}$, a quaternion that represents how the Gaussian rotates $\mathbf{q}_{\text{mlp}}$ and the ambient occlusion scaling factor s.

We obtain the final Gaussian position p_{obs} and orientation $\mathbf{q}_{\text{obs}}$ by applying this residual transformation to the LBS output. Importantly, to enable generalization the residual translation vector is defined in the canonical space and thus needs to be rotated with the orientation from LBS:

$$\mathbf{p}_{\text{obs}} = \mathbf{p}_{\text{lbs}} + (\mathbf{R}_t \mathbf{t}_{\text{mlp}}) \qquad \mathbf{q}_{\text{obs}} = \mathbf{q}_{\text{mlp}} \circ \mathbf{q}_{\text{lbs}}. \quad (4)$$

3.4. Image rendering

Once the parameters of Gaussians in the observation space have been computed, we render the image using the fast and differentiable Gaussian rasterizer from 3D-GS [16].

3.5. Training procedure

At each training step, given an image and its corresponding body pose, we transform the canonical Gaussians from the canonical space to the observation space, render the image and finally optimize parameters from the Gaussians and the MLP with gradient descent. We use the SMPL model to initialize the primitives. Gaussian centers are set to the vertices positions, from which we can import the skinning weights from the template. During training, the set of Gaussians is incrementally densified and pruned, following the heuristics proposed by 3D-GS. We describe below the optimization objective of our method, which combines reconstruction and regularization losses. As shown in section 4.4, regularization plays an important role in guiding our over-parametrized model to a solution that generalizes to novel body poses.

Reconstruction losses The main objective of the model is to reconstruct the training images. Using the segmentation mask, we set the background pixels from the ground-truth image in black. We use a L_1 loss $\mathcal{L}_{L_1}$, a D-SSIM [61] loss $\mathcal{L}_{\text{ssim}}$ and a perceptual loss [70] $\mathcal{L}_{\text{lpips}}$ with VGG [54] weights between rendered and groundtruth images.

Minimize MLP output: We want our deformations to rely on LBS as much as possible and expect the MLP to learn only local deformations. Thus, we restrict the MLP

Figure 3. **Visualization of MLP outputs.** From left to right: ground-truth image, rendered image, translation output t_{mlp} norm (lightest colors indicate largest translation vector) and ambient occlusion factor s (grey: no color modification, blue: darker colors, red: lighter color). We observe that our MLP learns to operate on the dynamic parts of garments.

outputs to be as small as possible. We define one regularization term for each output: $\mathcal{L}_{\text{trans}}$ controls the norm of the translation residual, $\mathcal{L}_{\text{rot}}$ pushes the rotation residual close to the identity quaternion q_{id}, and $\mathcal{L}_{\text{amb}}$ guides the ambient occlusion factor to stay close to 1:

$$\mathcal{L}_{\text{trans}} = \|t_{\text{mlp}}\|_2 \quad \mathcal{L}_{\text{rot}} = \|q_{\text{mlp}} - q_{\text{id}}\|_1 \quad \mathcal{L}_s = \|s-1\|_2.$$

Regularization of canonical positions: We encourage Gaussian positions to stay close to the SMPL mesh to avoid floating artifacts. Because our model represents the clothed body, we define a threshold τ_{pos} that represents the maximum distance between the skin and the clothes. We search the nearest vertex v_i from each Gaussian canonical position p_i and apply the following loss, that penalizes points that are further than the threshold:

$$\mathcal{L}_{\text{mesh}} = \sum_i ReLU(\|v_i - p_i\|_2 - \tau_{\text{pos}}).$$

Skinning weights weak supervision: Because we train on a limited amount of gestures, the training data can usually be fitted with skinning weights that do not generalize well. Consequently, we softly supervise the skinning weights of our Gaussians with those from SMPL, *i.e.*, the closer a Gaussian is from a vertex, the more similar its skinning weights need to be:

$$\mathcal{L}_{\text{skn}} = \sum_i \text{ReLU}(\|w(g_i) - w(v_i)\|_2 - \tau_{\text{skn}}\|v_i - p_i\|_2).$$

It should be noted that we do not backpropagate the gradient of the distance $\|v_i - p_i\|_2$ through this loss.

We sum all the reconstruction and regularization terms to obtain the final loss $\mathcal{L}$. The weights and hyperparameters used are given in the supplementary materials.

4. Experiments

In Section 4.3, we compare Human Gaussian Splatting against state-of-the-art human avatars on novel views and novel pose synthesis on public datasets. Then, we perform several ablation studies in Section 4.4 to show the benefit our design choices. This research has been conducted with public datasets only, which, to the best of our knowledge, have collected human subject data according to regulations. More results are shown in supplementary materials, including videos of novel pose synthesis and a demonstration of real-time rendering.

4.1. Implementation details

Our algorithm is implemented in the PyTorch framework. The LBS module is inspired by the SMPL-X repository [42]. We optimize the total training objectives using Adam optimizer with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. More implementation details are given in the supplementary materials.

4.2. Dataset and evaluation metrics

We validate our method on 3 datasets: THuman4 [72] is captured by 24 calibrated RGB cameras at 30 fps with an image resolution of 1330×1150. 3 different subjects are covered, each sequence ranging from 2500 to 5000 frames. DNA-Rendering [5] uses 60 cameras to capture a wide range of human motions and clothing. ZJU-Mocap [44] is obtained using 23 hardware-synchronized cameras, with 1024×1024 resolution. Similar to existing work, we utilize PSNR, SSIM, and LPIPS to evaluate both novel view and novel pose synthesis. Following PoseVocab [26], we also include FID metric [8] on the THuman4 dataset to measure the realism between rendered and ground truth data.

4.3. Comparison with state-of-the-art

Baselines We compare our approach with state-of-the-art methods: 1) PoseVocab [26] uses SDF-based volume rendering and joint-structured embeddings, 2) SLRF [72] defines hundreds of local NeRFs defined on SMPL surface, 3) TAVA [23] is a template-free NeRF approach with forward skinning weights field, 4) Ani-NeRF [43], uses a canonical NeRF with backward skinning, 5) ARAH [60] defines a canonical SDF and finds backward correspondences with joint root-finding and 6) DVA [50] uses forward deformation of articulated volumetric primitives.

4.3.1 Evaluation on THuman4 dataset

For this dataset, we carefully replicate experiments proposed by PoseVocab [26] on the "subject00" sequence. One of the 24 cameras is held out for the evaluation of novel view synthesis. For novel pose synthesis, the method is trained with the first 2000 frames and evaluated on the rest 500 frames. Quantitative results are given in Tab. 1, where the score of other methods is reported from PoseVocab [26]

Table 1. **Quantitative comparison on Thuman4 dataset.** We evaluate the performance on both novel view and novel pose synthesis, and time efficiency. Our method achieves the best performance on all the metrics and supports real-time rendering in the inference stage.

Method	Training poses				Novel poses				Efficiency	
	PSNR	SSIM	LPIPS	FID	PSNR	SSIM	LPIPS	FID	Render (s)	Training (h)
HuGS(Ours)	35.05	0.99	0.020	9.48	32.49	0.984	0.019	11.76	0.015	10
PoseVocab [26]	34.23	0.99	0.014	23.957	30.97	0.977	0.017	37.239	3	48+
SLRF [72]	25.27	0.97	0.024	44.49	26.15	0.969	0.024	110.651	5	25
TAVA [23]	23.93	0.97	0.029	75.46	26.61	0.968	0.032	99.947	-	-
Ani-NeRF [43]	23.19	0.97	0.033	85.45	22.53	0.964	0.034	102.233	1.09	12
ARAH [60]	22.02	0.96	0.033	74.30	21.77	0.958	0.037	77.840	10	36

and the efficiency metrics have been collected on each respective paper, except for TAVA that does not report rendering time and indicates training time as a limitation. Our method obtains better PSNR, SSIM, and FID scores than all the competitors. In terms of time efficiency, the training time of our method is on par with or better with the baselines, while being the only method to present a real-time rendering in the inference stage. This comparison shows that our method can achieve state-of-the-art performance on both novel view synthesis and novel pose animation while maintaining real-time rendering speed. We show a qualitative comparison against the best competitor PoseVocab in Figure 6. HuGS and PoseVocab exhibit different strengths and weaknesses. Our method shows more fidelity to the groundtruth image, for example the logo on the hoody or the head pose which is perfectly aligned. On the other hand, PoseVocab exhibits a smoother surface thanks to SDF formulation. Another drawback of PoseVocab are artefacts that appears outside of the body topology, due to failure in the inverse skinning process. Because we use forward deformation, such artefacts do not happen with our method.

4.3.2 Evaluation on DNA-Rendering dataset

We further evaluate our method on DNA-Rendering [5] that proposes more challenging subjects with loose clothing and complex textures. We use 48 cameras and 80 % of the video to train on 3 sequences. Novel view synthesis is evaluated on 12 held out cameras and novel poses on the held out last frames with all cameras. We compare HuGS to DVA [50], which also performs forward deformation of 3D primitives. Quantitative comparison is provided in Tab. 2 and renderings are shown in Fig. 4. The combination of complex textures, fast non-rigid motion and loose clothing make it very difficult for both methods to render details with high fidelity. Nonetheless, our method exhibits better results than DVA and can fit unusual topology and preserve it under novel poses. This is because we rely on the template only at initialization and are able to fit the canonical representation to an arbitrary shape, unlike other methods such as DVA that define primitives close to the template surface.

Table 2. **Comparison with DVA on DNA-Rendering.**

Seq	Method	Novel views			Novel poses		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
0165	HuGS	31.5	0.98	0.022	30.0	0.97	0.025
	DVA [50]	29.8	0.97	0.025	28.8	0.97	0.036
0166	HuGS	27.0	0.97	0.050	25.7	0.96	0.056
	DVA [50]	26.1	0.96	0.059	25.4	0.95	0.063
0206	HuGS	25.7	0.96	0.061	23.2	0.94	0.073
	DVA [50]	22.8	0.94	0.076	23.1	0.93	0.079

4.3.3 Evaluation on ZJU-MoCap dataset

To benchmark HuGS on ZJU-MoCap [44], we follow the setting of NeuralBody [44], i.e. use only 4 out of 21 cameras to train, and report results given by SLRF [72]. On this dataset, we train only for 50k iterations because we observe that textures degrade on novel poses with further training, due to the sparse camera setup. Table 3 presents the quantitative results. Our method achieves the second-best performance on novel views and the best performance on novel pose synthesis, where we obtain the highest PSNR and competitive SSIM. Moreover, our approach is notably more efficient for training and rendering compared to the baselines. These results shows that HuGS can generalize well in novel pose animation tasks, as shown in Fig. 5.

Table 3. **Results on ZJU-MoCap dataset.** We present the second-best performance on novel view synthesis while outperforming all the baselines on PSNR metric for novel poses.

	Novel Views		Novel Poses	
Method	**PSNR**	**SSIM**	**PSNR**	**SSIM**
HuGS(Ours)	26.58	0.934	23.69	0.896
SLRF [72]	28.32	0.953	23.61	0.905
Neural Body [44]	25.79	0.928	21.60	0.870
Ani-NeRF [43]	24.38	0.903	21.29	0.860

4.4. Ablation studies

Animate with a single transformation The core of our method consists in combining two transformations: LBS and a pose-dependent MLP. We study the performance of

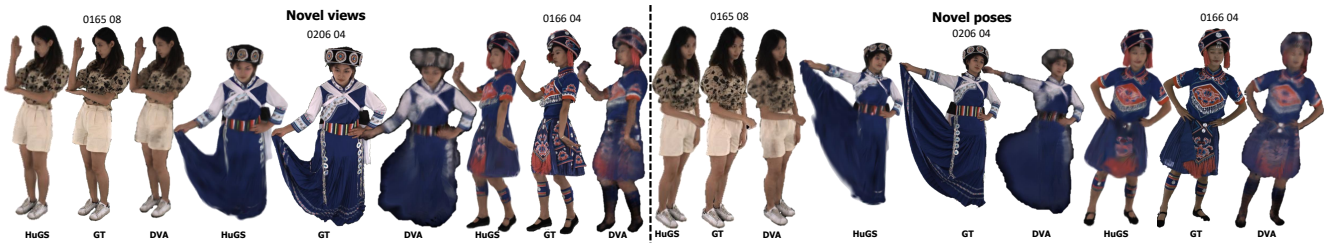

Figure 4. **Comparison with DVA on the DNA-Rendering dataset.** Despite fast non-rigid motion of complex textured garments, our method preserves more details than DVA and is able to fit unusual topology with loose clothing.

our method by using either LBS or MLP as the only deformation. Qualitative are shown in Fig. 7. Using only LBS is a strong baseline quantitatively competitive with the state-of-the-art. However, because it only encodes the rigid deformations of body joints, it is not able to model wrinkles on garments and lacks details compared to the proposed method. The MLP only version fails to model large deformations such that arms are not represented and thus is not a suitable algorithm for animatable avatars.

Learning skinning weights One contribution of our work is to learn per-gaussian skinning weights. We compare this design choice to a simpler baseline: a model where weights are not learned but imported from the closest SMPL vertex. Because the MLP can potentially correct errors from previous steps, we deactivate it for this experiment and deform gaussians only with LBS. We present the comparison in Tab. 4, where we observe that the model with learned skinning weights can bring more than 1dB improvement on PSNR over the SMPL skinning weights. The main reason is that SMPL weights are defined on the naked body while ours can adapt freely to any body shape. Template weights could also be inaccurate due to the pose estimation error in the training data, causing misalignement between the canonical gaussians and the template mesh.

Latent code or position We use a per-gaussian latent code as input of our MLP. It is in contrast with previous work [66] that used the position to identify the primitive. We train a model where we replace the latent code (16 floats) by canonical position augmented with fourier features [59] (63 floats). As shown in Tab. 5, latent codes perform slightly better than position while being more compact. With more MLP layers, we expect positional encodings to work similarly, but the learned features help to decode the information in small MLPs, similar to explicit features grids that accelerate NeRFs [38].

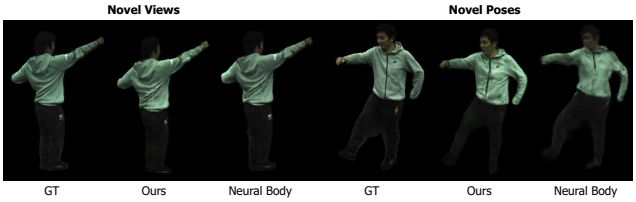

Figure 5. **Comparison with Neural Body on ZJU-MoCap dataset.** While results in novel pose synthesis are comparable for both methods, HuGS generalizes way better to novel poses thanks to its forward deformation formulation.

Table 4. **Ablation study on skinning weights.** We evaluated the ablated model with learned skinning weights and weights from the SMPL template on novel view synthesis.

Method	PSNR	SSIM	LPIPS	FID
Learned	31.99	0.984	0.020	27.15
Template	30.97	0.981	0.022	28.50

Table 5. **Ablation study on MLP input**. We compare learnable latent codes with canonical position encoding as MLP input, for novel pose synthesis on THuman4 Dataset.

Method	PSNR	SSIM	LPIPS	FID
Latent code	32.49	0.984	0.019	11.76
Position	32.20	0.985	0.019	17.78

Shading Finally, we verify the benefit of the shading component of our pipeline. We train a model where the MLP only outputs translation and rotation for each gaussian on the sequence00 of THuman4 dataset. This model obtains the following metrics on novel pose synthesis: 29.72dB PSNR, 0.978 SSIM, 0.027 LPIPS. These scores are directly comparable with those displayed in Tab 1. Removing this component forces the model to learn duplicate gaussians with different colors for shaded areas, leading to overfitting.

4.5. Efficiency

One of the main advantage of our method against previous work is its rendering time. We render an image of size 512x512 in $\approx$ 12ms or 80fps on a single Tesla V100

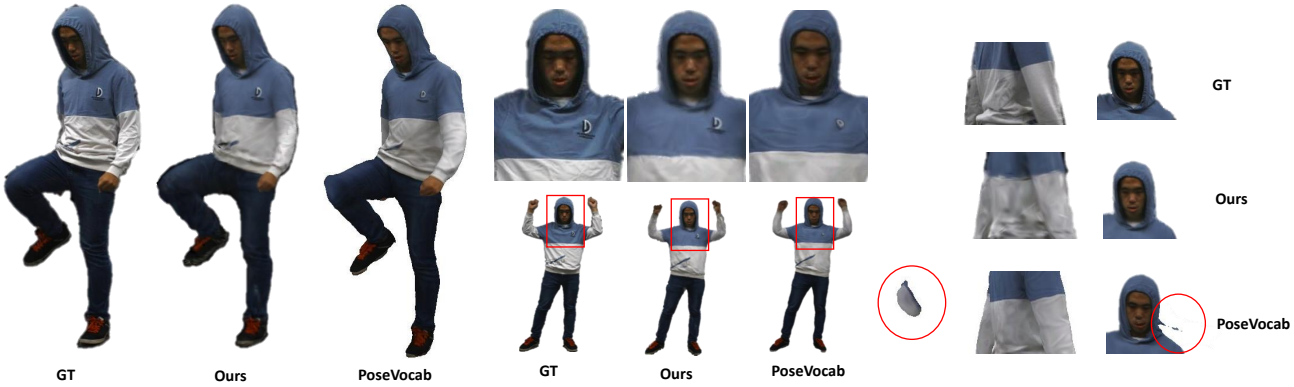

Figure 6. **Qualitative comparison between PoseVocab and HuGS on Thuman4 dataset.** On the left, our avatar shows better fidelity w.r.t. the body pose than PoseVocab that fails to deform the hood and the knee at the correct location. In the middle, HuGS presents a more detailed logo and more accurate head pose. On the right, PoseVocab presents artefacts due to failures of the inverse skinning process.

GPU. This runtime is two orders of magnitude faster than the compared SoTA, as shown in Table 1. Computing transformations from canonical to observation space takes 10ms and gaussians rasterization 2ms. Training the model for our experiments takes from 5 to 20 hours on a Tesla V100, depending on the dataset size. It compares equally or favorably to neural rendering competitors.

5. Discussion

Limitations and future work Our method is limited in several aspects: 1) the 3D reconstruction quality degrades with sparse camera setups because of overfitting on the training observations, resulting in poor novel pose synthesis. 2) The proposed MLP is able to fit observed garment deformations and replicate them for novel poses, but does not extrapolate novel deformations. 3) Each gaussian in our model is optimized and deformed independently, ignoring the relation between gaussians in local neighbourhoods. We think that defining structure and connectivity between primitives would help and leave it as future work.

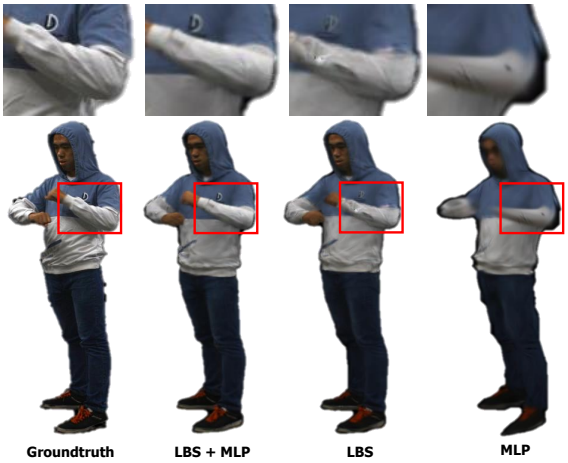

Figure 7. **Ablation study on the importance of combining LBS and MLP.** We show the images in the order of ground truth, qualitative results of our full model with both LBS and posed-based MLP, and the ablated model with LBS only and with MLP only.

Potential societal impacts The proposed algorithm could be used in Deep Fakes pipelines to synthesize fake videos of people with reenactment for malicious purpose. This aspect needs to be addressed and mitigated carefully.

Concurrent works There is a wide-spread interest in human modelling with gaussian splatting, such that many preprints with related approaches have been released during the review process of this paper [6, 9, 12–14, 17, 19, 21, 22, 27, 30, 39, 48, 49, 74]. We discuss key similarities and differences in the supplementary materials.

6. Conclusion

We have proposed **HuGS**, a first approach for creating and animating virtual human avatars based on Gaussian Splatting, by defining a coarse-to-fine deformation algorithm that combines forward skinning with local learning-based refinement. Using Gaussian Splatting for this problem not only helps to accelerate rendering, but also enables to bypass difficult inverse skinning approaches required in NeRF-based formulations. In contrast with other forward approaches, we are able to fit body shapes with loose clothing. Experimental results demonstrate that our approach can achieve state-of-the-art human neural rendering performance with good generalization. The fast rendering of this approach should facilitate its deployment and we hope that HuGS will also serve as an intuitive baseline for follow-up research on gaussian-based avatars.

References

[1] Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabian Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. Driving-signal aware full-body avatars. *ACM Transactions on Graphics (TOG)*, 40(4):1–17, 2021. 3

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 2

[3] Pol Caselles, Eduard Ramon, Jaime Garcia, Xavier Giro-i Nieto, Francesc Moreno-Noguer, and Gil Triginer. Sira: Relightable avatars from a single image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 775–784, 2023. 2

[4] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[5] Wei Cheng, Ruixiang Chen, Siming Fan, Wanqi Yin, Keyu Chen, Zhongang Cai, Jingbo Wang, Yang Gao, Zhengming Yu, Zhengyu Lin, et al. Dna-rendering: A diverse neural actor repository for high-fidelity human-centric rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19982–19993, 2023. 5, 6

[6] Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. *arXiv preprint arXiv:2312.02902*, 2023. 8

[7] Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Learning neural volumetric representations of dynamic humans in minutes. In *CVPR*, 2023. 2

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[9] 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. *arXiv preprint arXiv:2312.02134*, 2023. 8

[10] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2

[11] Youngkyoon Jang, Jiali Zheng, Jifei Song, Helisa Dhamo, Eduardo Pérez-Pellitero, Thomas Tanay, Matteo Maggioni, Richard Shaw, Sibi Catley-Chandar, Yiren Zhou, et al. Vschh 2023: A benchmark for the view synthesis challenge of human heads. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1121–1128, 2023. 2

[12] Rohit Jena, Ganesh Subramanian Iyer, Siddharth Choudhary, Brandon Smith, Pratik Chaudhari, and James Gee. Splatarmor: Articulated gaussian splatting for animatable humans from monocular rgb videos. *arXiv preprint arXiv:2311.10812*, 2023. 8, 1

[13] Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. Hifi4g: High-fidelity human performance rendering via compact gaussian splatting. *arXiv preprint arXiv:2312.03461*, 2023.

[14] HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Perez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3d gaussian splatting for animatable human avatars. *arXiv preprint arXiv:2312.15059*, 2023. 8, 1

[15] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007. 2

[16] 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, 3, 4, 1

[17] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats. *arXiv preprint arXiv:2311.17910*, 2023. 8

[18] Jonas Kulhanek and Torsten Sattler. Tetra-NeRF: Representing neural radiance fields using tetrahedra. *arXiv preprint arXiv:2304.09987*, 2023. 2

[19] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. *arXiv preprint arXiv:2311.16099*, 2023. 8, 1

[20] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000. 2

[21] Mingwei Li, Jiachen Tao, Zongxin Yang, and Yi Yang. Human101: Training 100+fps human gaussians in 100s from 1 view, 2023. 8, 1

[22] Mengtian Li, Shengxiang Yao, Zhifeng Xie, Keyu Chen, and Yu-Gang Jiang. Gaussianbody: Clothed human reconstruction via 3d gaussian splatting. *arXiv preprint arXiv:2401.09720*, 2024. 8

[23] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 5, 6

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

[25] Zhuopeng Li, Lu Li, and Jianke Zhu. Read: Large-scale neural scene rendering for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1522–1529, 2023. 2

[26] Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. Posevocab: Learning joint-structured pose embeddings for human avatar modeling. In *ACM SIGGRAPH Conference Proceedings*, 2023. 2, 3, 5, 6

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

for high-fidelity human avatar modeling. *arXiv preprint arXiv:2311.16096*, 2023. 8, 1

[28] Tingting Liao, Xiaomei Zhang, Yuliang Xiu, Hongwei Yi, Xudong Liu, Guo-Jun Qi, Yong Zhang, Xuan Wang, Xiangyu Zhu, and Zhen Lei. High-Fidelity Clothed Avatar Reconstruction from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[29] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.(ACM SIGGRAPH Asia)*, 2021. 2

[30] Yang Liu, Xiang Huang, Minghan Qin, Qinwei Lin, and Haoqian Wang. Animatable 3d gaussian: Fast and high-quality reconstruction of multiple human avatars. *arXiv preprint arXiv:2311.16482*, 2023. 8

[31] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Trans. Graph.*, 40(4), 2021. 3

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2, 3

[33] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2

[34] Àlex Méndez-Feliu and Mateu Sbert. From obscurances to ambient occlusion: A survey. *The Visual Computer*, 25:181–196, 2009. 4

[35] Bruce Merry, Patrick Marais, and James Gain. Animation space: A truly linear framework for character animation. *ACM Transactions on Graphics (TOG)*, 25(4):1400–1423, 2006. 2

[36] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[37] Ansh Mittal. Neural radiance fields: Past, present, and future. *arXiv preprint arXiv:2304.10050*, 2023. 2

[38] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2, 7

[39] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. Ash: Animatable gaussian splats for efficient and photoreal human rendering. *arXiv preprint arXiv:2312.05941*, 2023. 8, 1

[40] 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. 1

[41] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[42] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 2, 3, 5, 1

[43] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021. 2, 3, 5, 6

[44] 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. In *CVPR*, 2021. 2, 5, 6

[45] Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Representing volumetric videos as dynamic mlp maps. In *CVPR*, 2023. 2

[46] Sergey Prokudin, Qianli Ma, Maxime Raafat, Julien Valentin, and Siyu Tang. Dynamic point fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7964–7976, 2023. 3

[47] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[48] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv preprint arXiv:2312.02069*, 2023. 8

[49] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. *arXiv preprint arXiv:2312.09228*, 2023. 8

[50] Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, et al. Drivable volumetric avatars using texel-aligned features. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 3, 5, 6

[51] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3

[52] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[53] Richard Shaw, Jifei Song, Arthur Moreau, Michal Nazarczuk, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Perez-Pellitero. Swags: Sampling windows adaptively for dynamic 3d gaussian splatting. *arXiv preprint arXiv:2312.13308*, 2023. 2

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[55] Shih-Yang Su, Timur Bagautdinov, and Helge Rhodin. Npc: Neural point characters from video. In *ICCV*, 2023. 3

[56] Zhaoqi Su, Liangxiao Hu, Siyou Lin, Hongwen Zhang, Shengping Zhang, Justus Thies, and Yebin Liu. Caphy: Capturing physical properties for animatable human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14150–14160, 2023. 2, 3

[57] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2

[58] Andrea Tagliasacchi and Ben Mildenhall. Volume rendering digest (for nerf). *arXiv preprint arXiv:2209.02417*, 2022. 2

[59] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020. 7

[60] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *European Conference on Computer Vision*, 2022. 2, 3, 5, 6

[61] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4

[62] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, 2022. 2

[63] 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. *arXiv preprint arXiv:2310.08528*, 2023. 2

[64] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. 2

[65] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 2

[66] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2, 7

[67] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv 2310.10642*, 2023. 2

[68] Haitao Yu, Deheng Zhang, Peiyuan Xie, and Tianyi Zhang. Point-based radiance fields for controllable human motion synthesis. *arXiv preprint arXiv:2310.03375*, 2023. 3

[69] Zhengming Yu, Wei Cheng, xian Liu, Wayne Wu, and Kwan-Yee Lin. MonoHuman: Animatable human neural field from monocular video. In *CVPR*, 2023. 2

[70] 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. 4

[71] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[72] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 5, 6, 1

[73] Zerong Zheng, Xiaochen Zhao, Hongwen Zhang, Boning Liu, and Yebin Liu. Avatarrex: Real-time expressive full-body avatars. *ACM Transactions on Graphics (TOG)*, 42(4), 2023. 2

[74] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. *arXiv preprint arXiv:2311.08581*, 2023. 8

Human Gaussian Splatting: Real-time Rendering of Animatable Avatars

Supplementary Material

We present further analysis of our method. We invite readers to watch the video that summarizes our contributions and demonstrates real-time rendering, whose details are given in section 7. We further present extensive implementation details and hyperparameters setting in section 8, in order to facilitate the reproducibility of our experiments. Finally, we provide more qualitative results of our method on the THuman4 dataset in section 10.

7. Real-time video rendering

Real-time rendering in the supplementary video The attached video showcases real-time novel pose synthesis on the THuman4 dataset [72] at 60fps. This is done by extending the viewer from 3D-GS to dynamic scenarios in order to render videos.

8. Reproducibility details

We have described our main implementation details in the main manuscript. In this section, we further report the hyperparameter values of our pipeline in Table 6. After that, we provide the additional implementation details of our approach, as described as below.

Linear blend skinning Our implementation for linear blend skinning (LBS) follows SMPL-X [42]. Notably, we do not use pose blend shapes before applying deformations on human joints, because the shape estimation and blend shapes obtained upon that may be inaccurate, thus we design MLP to handle pose-dependent deformations. We also remind that in our case, LBS is applied on canonical gaussians only and thus deforming template vertices is not necessary. The learnable per-gaussian skinning weights vector $\mathbf{w}$ is a parameter optimized through gradient descent which can leads to negative values. We apply a ReLU activation on each $\mathbf{w}_j$ and then normalize the vector such that its components sum to 1 to obtain a well defined skinning weights vector. Finally, the transformations matrices $\mathbf{M}_{j,t}$ that encode the rigid deformation of each body joint j for each training timestep t are precomputed before training to improve efficiency.

Learning rates Similar to 3D-GS [16], we use different learning rates for each set of learnable parameters. We leave the learning rates of the original parameters (position, orientation, scaling, colors and opacity) unchanged. Our MLP and the skinning weights vectors $\mathbf{w}$ are optimized with a constant learning rate that is set to $1e^{-4}$. For latent codes $\mathbf{l}$, we use a learning rate of $2.5e^{-3}$.

Table 6. **Hyperparameters values.**

Parameter name	Value
λ_{L_1} (in Eqn. 8)	0.8
λ_{ssim} (in Eqn. 8)	0.2
λ_{lpips} (in Eqn. 8)	0.05
λ_{trans} (in Eqn. 8)	0.01
λ_{rot} (in Eqn. 8)	0.001
λ_{s} (in Eqn. 8)	0.001
λ_{mesh} (in Eqn. 8)	0.1
λ_{skn} (in Eqn. 8)	0.001
Dimension of latent code $\mathbf{l}$	16

9. Concurrent works

Gaussian splatting is currently a very active research topic and while this work was under review, many related drivable avatars based on gaussian splatting have been released. We propose a small discussion and refer to Awesome 3D Gaussian splatting for a complete list of related papers.

Similar to HuGS, most gaussian avatars exploit forward deformation of gaussians with LBS to drive the avatar. Local refinement with a neural network is also a popular choice [12, 14, 21, 27] but different designs have been developed. Notably, SplatArmor [12] uses canonical gaussian parameters as input, Animatable Gaussians [27] and ASH [39] use a 2D CNN. The per-gaussian latent code and the shading components proposed by our method are the main advantages of HuGS compared to these approaches. Regarding skinning weights, most methods rely on the template, with the exception of GART [19] that also optimizes these parameters. In contrast with ours, these learnable skinning weights are not defined per-gaussian but in a voxel grid. Finally, similar to ours, most methods optimize the canonical gaussians jointly with the rest of the pipeline. In contrast, Animatable Gaussians [27] and ASH [39] import a template (such as a SDF) where the body shape as already been fitted on the subject. This design choice adds an expensive pre-processing step but also seems to exhibit very good results. We expect follow-up research to build on this large amount of proposals to push gaussian-based animatable avatars forward.

10. Qualitative analysis

We display in Figure 8 qualitative results of the HuGS method for subject01 and subject02 sequences from the THuman4 dataset [72] for novel pose synthesis. Note that no quantitative comparison is done on these subjects be-

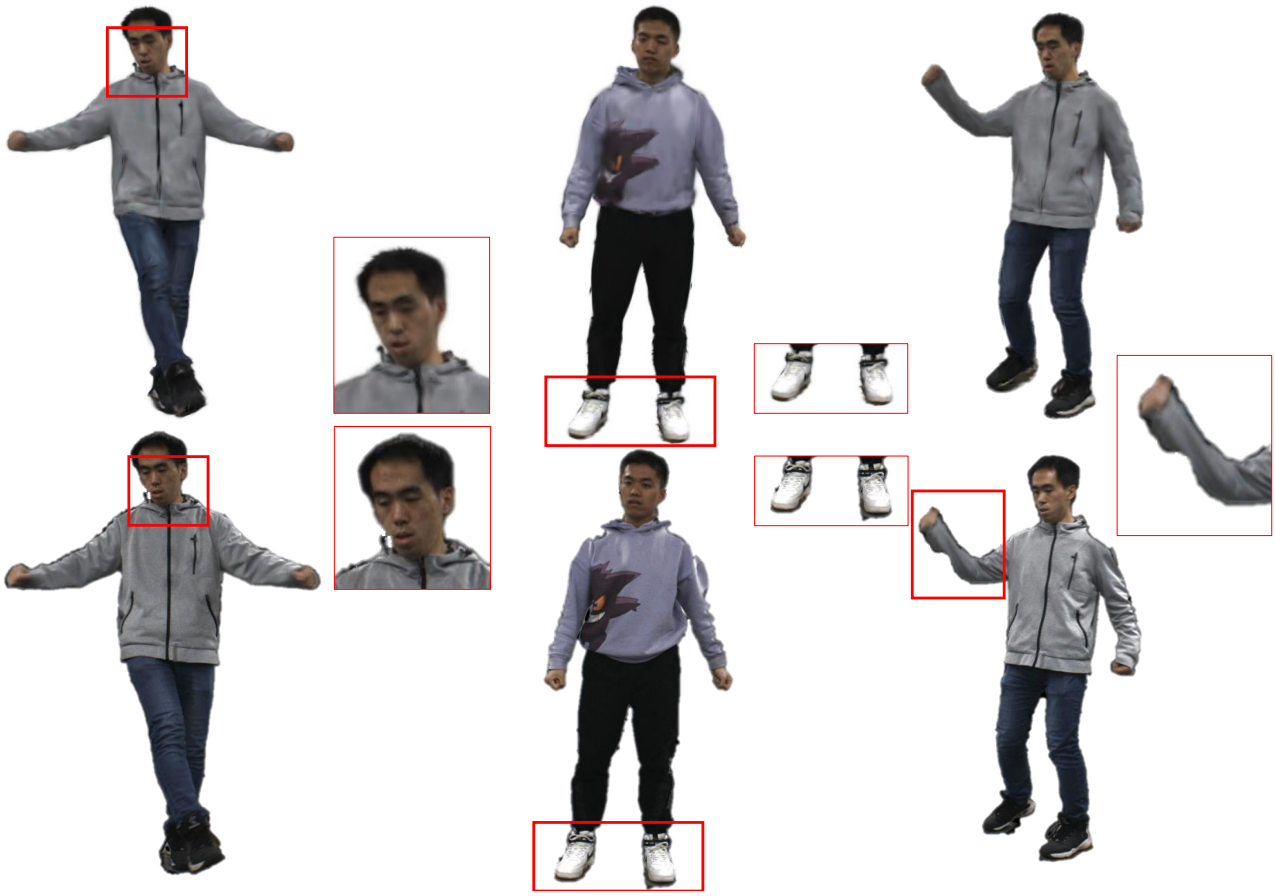

Figure 8. **Qualitative visualization of HuGS novel pose synthesis on THuman4 dataset.**

cause the evaluation setup has not been released by the dataset authors. We observe that our method is able to fit the subjects with precise details, such as the black hood button (left picture) or the shoes (middle), and render the target body pose with high fidelity. However, we also showcase inaccuracies in the dataset caused by segmentation masks and motion blur that are observed regularly on training images and thus create artifacts in the learned model and degrade the overall rendering quality on these subjects.