Animatable 3D Gaussians for Modeling Dynamic Humans

ANONYMOUS AUTHOR(S)
SUBMISSION ID: 528

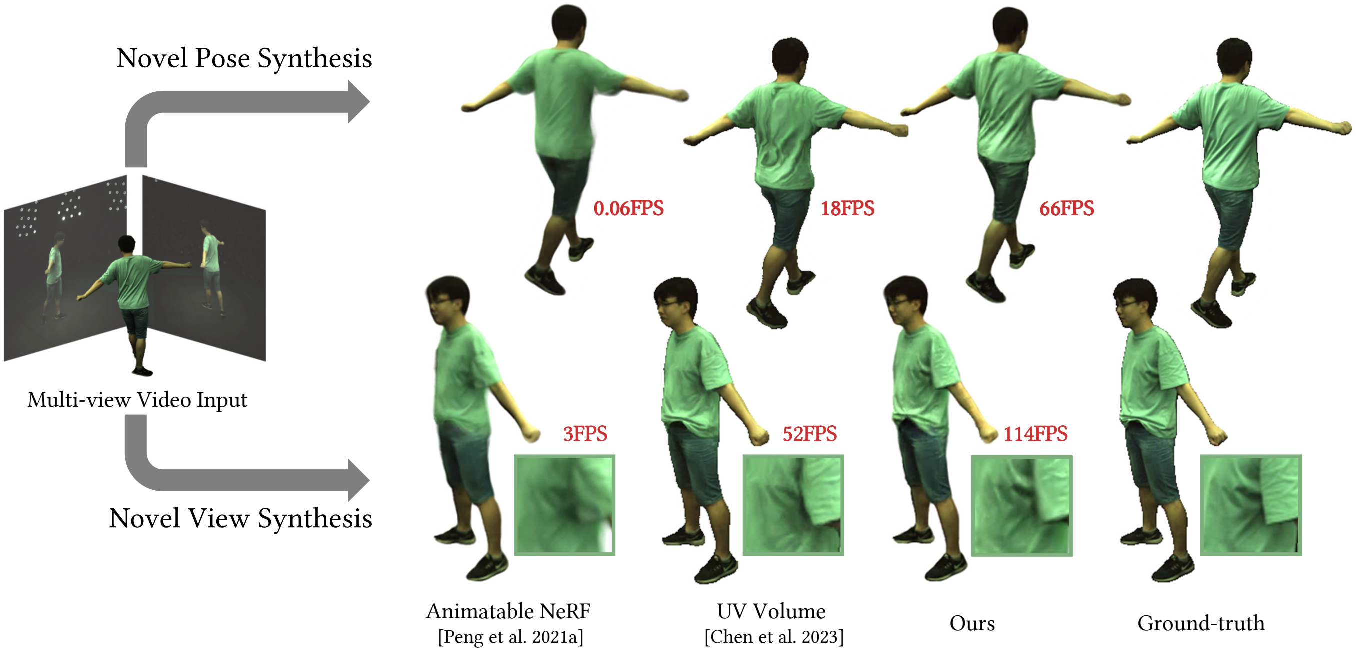

Fig. 1. Compared with state-of-the-art techniques, our animatable 3D Gaussian representation is able to capture high-frequency details and achieve superior rendering performance.

We present an animatable 3D Gaussian representation for synthesizing high-fidelity human videos under novel views and poses in real time. Given multi-view videos of a human subject, we learn a collection of 3D Gaussians in the canonical space of the rest pose. Each Gaussian is associated with a few basic properties (i.e., position, opacity, scale, rotation, spherical harmonics coefficients) representing the average human appearance across all video frames, as well as a latent code and a set of blend weights for dynamic appearance correction and pose transformation. The latent code is fed to an Multi-layer Perceptron (MLP) with a target pose to correct Gaussians in the canonical space to capture appearance changes under the target pose. The corrected Gaussians are then transformed to the target pose using linear blend skinning (LBS) with their blend weights. High-fidelity human images under novel views and poses can be rendered in real time through Gaussian splatting. Compared to state-of-the-art NeRF-based methods, our animatable Gaussian representation produces more compelling results with well captured details, and achieves superior rendering performance.

CCS Concepts: • **Computing methodologies** → **Rasterization**; **Point-based models**; **Machine learning approaches**; *Rendering*.

Additional Key Words and Phrases: free-view videos, image-based rendering, Gaussian splatting

ACM Reference Format:
Anonymous Author(s). 2024. Animatable 3D Gaussians for Modeling Dynamic Humans. *ACM Trans. Graph.* 1, 1 (January 2024), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Synthesizing photorealistic human animations constitutes a critical challenge across various domains, including telepresence, free-view videos, and cinematography. Conventional methods [Allen et al. 2003; Loper et al. 2023] apply 3D mesh reconstruction for this task. The reconstructed meshes, however, may not capture complex geometry details well, leading to noticeable degradation in visual quality. Neural radiance field (NeRF) [Mildenhall et al. 2020] offers a new perspective to 3D representation, which encodes the color and geometry information of a 3D scene with an Multi-layer Perceptron (MLP) network, and performs rendering via volumetric ray-marching. Recent work [Chen et al. 2023; Peng et al. 2021a; Zhao et al. 2022] has successfully applied NeRF to dynamic human modeling and demonstrated promising results in free view synthesis.

Nonetheless, discernible artifacts persist in NeRF-generated human videos. Notably, these techniques often manifest blurred results and cannot capture high-frequency details exhibited in input video

ACM 0730-0301/2024/1-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

frames (e.g., garment wrinkles) [Peng et al. 2021a,b]. State-of-the-art methods [Chen et al. 2023] propose to enhance the NeRF representation with a learned UV texture generator to produce intricate human details. However, the generated textures could be inconsistent across different human poses, resulting in noticeable jittering artifacts in synthesized videos (see the supplementary video). NeRF-based methods also have high computational costs, making it difficult to realize real-time synthesis of animated humans (with the exception of [Chen et al. 2023; Lin et al. 2022]).

In this paper, we propose an animatable 3D Gaussian representation for synthesizing high-fidelity human videos under novel views and poses in real time. Compared with NeRF-based methods, 3D Gaussian splatting (3DGS) [Kerbl et al. 2023] provides a competitive solution to novel view synthesis in rendering high-resolution images at real-time frame rates. However, extending 3DGS to model animatable humans is non-trivial – the original method is designed for static scenes. While recent concurrent work [Yang et al. 2023] has demonstrated dynamic scene modeling using Gaussians, it is restricted to video replay and not suitable for synthesizing dynamic humans under novel views and poses.

Our animatable Gaussian representation leverages multi-view human videos as input, and learns a collection of 3D Gaussians in the canonical space of the rest pose. Each Gaussian is associated with a few basic properties (i.e., position, opacity, scale, rotation, spherical harmonics coefficients), along with a latent code and a set of blend weights. The Gaussians with basic properties represent the *average* human appearance across all video frames. The latent code serves as a pose-aware residual appearance embedding. Given a target pose, the latent code and the target pose are fed to a tiny MLP to correct each Gaussian in the canonical space to capture the appearance changes under the target pose. The corrected Gaussians are then transformed to the target pose using linear blend skinning (LBS) [Jacobson et al. 2014] with their blend weights. In this way, high-fidelity human images under novel views and poses can be rendered in real time using Gaussian splatting.

To learn the animatable Gaussian representation, we elaborate several loss function terms including the image loss, D-SSIM loss, and perceptual loss that are commonly used in prior work, as well as a blend weight loss and an alpha loss dedicated for our representation. The blend weight loss is introduced to suppress the standard deviation of blend weights within each Gaussian, which ensures that each Gaussian can undergo a LBS transformation as a cohesive unit without introducing significant errors. The LBS transformation establishes a continuous deformation field in 3D space by employing a linear combination of bone matrices with associated blend weights at any 3D point. However, in our representation, each Gaussian can only be transformed as a cohesive unit, using the transformation at its center. Consequently, the transformation of Gaussians is essentially an approximation of the continuous LBS transformation. If the blend weights within a Gaussian vary greatly, the approximation will result in substantial errors, manifesting noticeable artifacts such as Gaussians protruding outside the human body (see Fig. 6). The alpha loss is defined as the deviation of the rendered opacity image from the binary foreground mask image, which explicitly constrains Gaussians to stay within the human region without the interference of the background, and makes Gaussians better capture the movement of garments.

We conduct a two-stage approach to train the Gaussian representation. In the first stage, we only optimize the basic Gaussian properties and human joint parameters to obtain an average human model as an initial configuration. In the second stage, we switch on the MLPs to empower Gaussians in capturing pose-aware appearance changes and acquiring more precise blend weights. Such a two-stage scheme effectively improve the robustness of the training procedure (see Fig. 6).

Based on the animatable Gaussian representation, we can synthesize high-quality free-view human videos in novel poses. Compared to previous NeRF-based methods, our method can better capture high-frequency details, which are consistent across different poses, producing temporally stable human videos. As we only need a tiny MLP for Gaussian correction at runtime, and thanks to the superior rendering performance of Gaussian splatting, animated human synthesis can be performed in real time, significantly faster than state-of-the-art techniques (66 fps versus 18 fps in [Chen et al. 2023] in novel pose synthesis). We conduct extensive experiments on three established datasets: ZJU Mocap, H36M, and CMU Panoptic datasets. Both qualitative and quantitative results show the superiority of our method over existing techniques.

2 RELATED WORK

Free-View Human Video Synthesis. In the last decade, many efforts have been made to model dynamic humans. Some work attempts to build a statistical mesh template [Joo et al. 2018; Loper et al. 2023; Osman et al. 2020] to model human bodies. To handle human appearance, traditional methods scan human subjects to acquire textures and material parameters [Allen et al. 2003; Zhang et al. 2017]. For the deformable parts such as loose garments, physical simulation [Guan et al. 2012], blending from database [Xu et al. 2011], or deformation space modeling [Habermann et al. 2021] are performed to improve fidelity. In recent years, lots of works leverage neural representations to depict dynamic scenes or humans, including voxels [Lombardi et al. 2019], point clouds [Wu et al. 2020], neural textures [Bagautdinov et al. 2021; Ma et al. 2021], and NeRF [Peng et al. 2021a; Xu et al. 2023a,b; Yang et al. 2022; Zhao et al. 2022]. Animatable NeRF [Peng et al. 2021a] uses the skinned multi-person linear model (SMPL) [Loper et al. 2023] to establish correspondences between arbitrary poses and the rest pose, and model pose-dependent details by conditioning an MLP on the appearance latent code of each frame. To model more local details, Zheng et al. [2022] assemble the radiance field of dynamic humans by a set of local ones, which improves the visual quality of garment wrinkles. However, these methods based on neural representation suffer from slow training and rendering. Fourier PlenOctrees [Wang et al. 2022] utilizes Fourier transformation to compact the dynamic octrees of the scene in the time domain, which realizes 100fps rendering but does not support novel pose generation. InstantAvatar [Jiang et al. 2023] incorporates instant-NGP [Müller et al. 2022] in avatar learning from monocular video input, and achieves 15fps rendering performance. As NeRF-based methods tend to generate blurred results, UV Volume [Chen et al. 2023] proposes to render a UV map

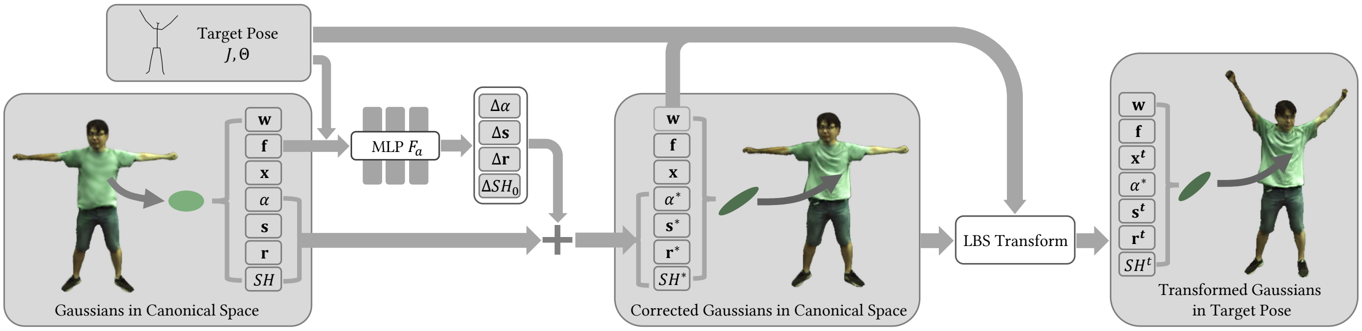

Fig. 2. Overview of our method. The animatable 3D Gaussian representation learns a collection of 3D Gaussians in the canonical space of the rest pose from multi-view human videos. Each Gaussian is associated with the position $\mathbf{x}$, opacity α, scale $\mathbf{s}$, rotation $\mathbf{r}$, spherical harmonics SH, along with the blend weights $\mathbf{w}$ and a latent code $\mathbf{f}$. For a given target pose, the latent code and pose are fed to an MLP F_a to correct each Gaussian in the canonical space to capture the appearance changes under the target pose. The corrected Gaussians are then transformed to the target pose using LBS with their blend weights.

by neural volume rendering and uses a generator to obtain textures conditioned on the pose. UV Volume can render appearance with high-frequency details and achieve real-time rendering, but its texture prediction is inconsistent across video frames, causing jittering artifacts in synthesized videos. Different from NeRF-based techniques, our method applies the SMPL model to explicitly transform the 3D Guassians and distributes an appearance latent code to each Gaussian, which is decoded by a tiny MLP to obtain pose-dependent human appearances. Our method produces high quality videos and keeps real-time rendering performance.

Novel View Synthesis for Static Scenes. Novel view synthesis for static scenes is a well-explored problem, which aims to synthesize new images from arbitrary views of a scene. Traditional approaches [Buehler et al. 2023; Davis et al. 2012; Eisemann et al. 2008] construct light fields to generate novel views from densely captured images. Recently, Neural Radiance Field (NeRF) [Mildenhall et al. 2020] has became a popular technique for this task, by representing the scene with implicit fields of view-dependent color and density using deep MLPs. Although NeRF achieves high-quality novel views, its training and rendering are time-consuming. Subsequent work [Garbin et al. 2021; Müller et al. 2022; Yu et al. 2021] attempts different strategies to accelerate NeRF. For example, instant-NGP [Müller et al. 2022] replaces the deep MLP with a shallow MLP, using multi-resolution hash encoding as its input, which can be trained in a few minutes and render images in real time. Recently, 3D Gaussian splatting (3DGS) [Kerbl et al. 2023] has demonstrated the superiority of explicit representations in novel view synthesis tasks. 3DGS builds a differentiable rasterizer to optimize the position, covariance and appearance of 3D Gaussians from image loss. Compared to NeRF-based methods which rely on expensive volumetric ray marching, 3DGS utilizes the traditional rasterization pipeline, achieving over 100 fps rendering. In addition, the explicit representation provides a more intuitive way for animation control, which motivates us to apply 3D Gaussians for modeling dynamic humans.

Concurrent works. Many concurrent works propose to model dynamic humans using 3D Gaussians. Zielonka et al. [2023] embed 3D Gaussians within human tetrahedral cages and employ cage deformations to model the pose-dependent variations. Each Gaussian is confined within a cage, and the total number of Gaussians remains fixed during optimization, which limits its capability to capture high-frequency details. Li et al. [2023] extract the color of each Gaussian from the Gaussian map predicted by the StyleUNet [Wang et al. 2023], which limits their rendering speed. Along with [Jena et al. 2023; Moreau et al. 2023],these works fail to achieve fast training with relatively complex pipelines. Kocabas et al. [2023] parameterize human Gaussians by their mean locations in a canonical space and their features from a triplane, but they do not take pose-dependent cloth deformation into account. Some other methods [Hu and Liu 2023; Lei et al. 2023] ignore pose-dependent fine details to achieve faster training and rendering, while our method strikes a good balance between realistic rendering and real-time performance. [Hu et al. 2023; Jena et al. 2023; Xiang et al. 2023] focus on modeling dynamic humans from monocular videos. They utilize additional regularization strategies, such as using MLPs to compute Gaussian colors to mitigate overfitting, which are orthogonal to our work.

3 METHOD

3.1 Overview

Our approach takes multi-view human videos as input. Following NeRF-based methods [Peng et al. 2021a], we extract foreground human masks [Lin et al. 2021], as well as 3D human poses (i.e., joint rotations and positions) [eas 2021], and 3D human bodies (SMPL) [Loper et al. 2023] from the videos.

The overview of our animatable 3D Gaussian representation is illustrated in Fig. 2. It includes a collection of 3D Gaussians in the canonical space of the rest pose (Sec. 3.2). Each Gaussian possesses a few basic properties representing the *average* human appearance across all video frames (Fig. 2 left), a latent code for Gaussian correction in the canonical space to reflect the appearance changes under a novel pose (Fig. 2 middle), and a set of blend weights for transforming the corrected Gaussians to the target pose using LBS (Fig. 2 right). We will discuss how to learn the representation in Sec. 3.3 and Sec. 3.4.

3.2 Animatable 3D Gaussians

We learn a collection of 3D Gaussians $\{G_1, G_2, .., G_N\}$ in the canonical space from the input videos. Each Gaussian is associated with a few basic properties (i.e., position $\mathbf{x}_i$, opacity α_i, anisotropic scale $\mathbf{s}_i$, rotation $\mathbf{r}_i$, spherical harmonics coefficients SH_i), along with a learnable code $\mathbf{f}_i$ and a set of blend weights $\mathbf{w}_i$. The Gaussians with basic properties represent the *average* human appearance in the canonical space across all video frames. The latent code $\mathbf{f}_i$ serves as a pose-aware residual appearance embedding, which is fed to a Gaussian correction model with a target pose, to correct the Gaussians in the canonical space to reflect the appearance change under the target pose.

Specifically, given a latent code $\mathbf{f}_i$ and a target pose $\Theta \in \mathbb{R}^{K\times 3}$ represented as rotations of K joints, the Gaussian correction model is defined as an MLP F_a:

$$\{\Delta\alpha_i, \Delta\mathbf{s}_i, \Delta\mathbf{r}_i, \Delta SH_{i0}\} = F_a(\Theta, \mathbf{f}_i). \tag{1}$$

The Gaussian properties are corrected accordingly (with position unchanged) as

$$\begin{gathered}\alpha_i^* = \alpha_i + \Delta\alpha_i, \mathbf{s}_i^* = \mathbf{s}_i + \Delta\mathbf{s}_i, \mathbf{r}_i^* = \Delta\mathbf{r}_i\mathbf{r}_i,\\ SH_i^* = \{SH_{i0} + \Delta SH_{i0}, SH_{i1}, SH_{i2}, SH_{i3}\}.\end{gathered} \tag{2}$$

Note that we only correct the zero-order component of spherical harmonics (i.e., the base or diffuse color). We find that optimizing higher-order components (i.e., the view-independent color) may cause ambiguity between pose-dependent and view-dependent variations, leading to degraded result in novel view and pose synthesis.

After Gaussian correction in the canonical space, we can transform the corrected Gaussians to a target pose. We utilize the SMPL human model [Loper et al. 2023] for this task. The human body has K parts with K transformation matrices $\mathbf{P}_k \in SE(3)$ (computed from the joint rotations Θ_k and joint positions J_k). For each Gaussian G_i, its corresponding transformation is

$$\mathbf{P}_i = \sum_{k=1}^{K} w_{ik}\mathbf{P}_k, \tag{3}$$

where $\mathbf{w}_i = \{w_{i1}, w_{i2}, ..., w_{iK}\}$ are the learned blend weights stored with each Gaussian. For Gaussian position $\mathbf{x}_i$, we find its closest point and corresponding triangle on the SMPL surface, and obtain the initial blend weights $\mathbf{w}_i^0$ using barycentric interpolation of the weights of triangle vertices. As Gaussian positions keep changing during training, for computation efficiency, we follow the same strategy as [Peng et al. 2021a], that is, precomputing weights on a dense grid and computing weights using interpolation in the grid during the training. The initial weights may be inaccurate for Gaussians that are far way from the SMPL body and represent garments. We further apply a positional encoded MLP network to predict residual weights $F_{\Delta\mathbf{w}} : \mathbf{x} \longrightarrow \Delta\mathbf{w}(\mathbf{x})$, and the final blend weights are computed as $\mathbf{w}_\mathbf{i} = \mathbf{w}_i^0 + F_{\Delta\mathbf{w}}(\mathbf{x}_\mathbf{i})$. Note the MLP $F_{\Delta w}$ is only required in the training stage. After training, the final $\mathbf{w}_\mathbf{i}$ is stored with each Gaussian and directly fetched at runtime.

The transformation $\mathbf{P}_i$ for each Gaussian is decomposed to scaling $\mathbf{S}_i$, rotation $\mathbf{R}_i$ and translation $\mathbf{T}_i$ [Shoemake and Duff 1992]. We omit the shear component to prevent Gaussians from distortion. The Gaussians are transformed as

$$\begin{gathered}\mathbf{x}_i^t = \mathbf{R}_i\mathbf{x}_i^* + \mathbf{T}_i, \mathbf{s}_i^t = \mathbf{S}_i \odot \mathbf{s}_i^*, \mathbf{r}_i^t = \mathbf{R}_i\mathbf{r}_i^*,\\ SH_i^t = SH\_Rotation(\mathbf{R}_i, SH_i^*),\end{gathered} \tag{4}$$

where $\odot$ represents element-wise multiplication.

The transformed Gaussians $\{G_i^t\}$ are finally rendered to produce high-quality human images under novel views and poses. We apply the same rasterizer as [Kerbl et al. 2023] to perform differentiable rendering and obtain the rendered image I_{render}.

3.3 Training

From the animatable Gaussian representation, we render the human image for the particular pose and view of each input video frame to perform training. We jointly optimize the basic Gaussian properties, latent codes $\{\mathbf{f}_i\}$, as well as the MLP parameters of F_a and $F_{\Delta w}$. Noticing that the joint rotations Θ and joint positions J estimated from SMPL may not be accurate, we also optimize Θ and J during training.

The training aims to minimize the following loss function with five terms:

$$L = \lambda_1 L_{rgb} + \lambda_2 L_\alpha + \lambda_3 L_w + \lambda_4 L_{D-SSIM} + \lambda_5 L_p, \tag{5}$$

where $\lambda_1 = 0.8, \lambda_2 = 10, \lambda_3 = 0.2, \lambda_4 = 0.2, \lambda_5 = 0.2$ in all our tests.

L_{rgb} is the image loss by measuring the L_1 difference between the rendered images I_{render} and the video frames I_{gt}. We make use of a human boundary mask M_b when computing the image loss. The boundary mask sets pixels n-pixel away from the human boundary 0 while all other pixels 1 ($n = 5$ in our tests). We find that this simple approach effectively prevent Gaussians from fitting the zigzags around the boundary. The image loss is computed as:

$$L_{rgb} = \sum_{j=1}^{F} (||(I_{render}^j - I_{gt}^j) \odot M_b||), \tag{6}$$

where F is the frame number and $\odot$ is the pixelwise multiplication.

In order to alleviate the background interference during training, a simple scheme is to only compute the image loss on the human region defined by the foreground mask M_h. However, we find this scheme cannot prevent the Gaussians from growing out of the human region. To overcome this problem, we design an alpha loss L_α to explicitly constrain Gaussians to stay within the human region, by comparing the rendered opacity image with the foreground mask M_h. Specifically, we set all Gaussian colors to pure white and perform Gaussian splatting to obtain the accumulated opacity image $I_{opacity}$. The alpha loss L_α is thus defined as:

$$L_\alpha = \sum_{j=1}^{F} (||(I_{opacity}^j - M_h^j) \odot M_b||_2). \tag{7}$$

L_w is a blend weight loss to ensure that each Gaussian can undergo a LBS transformation as a cohesive unit without introducing significant errors. The LBS transformation establishes a continuous deformation field in 3D space, while in our representation each Gaussian is transformed as a cohesive unit, using the transformation of its center. Therefore the transformation of Gaussians is an approximation of the continuous LBS transformation. If the blend weights vary greatly within a Gaussian, such approximation will

Table 1. Quantitative results of novel view synthesis. Our method outperforms baselines (AN [Peng et al. 2021a] and UV [Chen et al. 2023]) on PSNR and SSIM and present competitive LPIPS. After adding the perceptual loss L_p, our method also achieves the best LPIPS.

Datasets		ZJU						H36M						CMU		
		313	315	386	387	390	392	s1p	s5p	s6p	s7p	s8p	s9p	p1	p4	p6
PSNR ↑	AN	30.42	28.61	34.33	31.11	35.47	32.54	28.66	29.67	29.42	29.51	28.12	29.44	32.43	30.01	30.39
	UV	32.20	28.87	36.61	31.16	36.26	32.49	28.80	30.12	29.50	29.49	28.50	29.25	32.94	31.95	31.76
	Ours, w/ L_p	33.70	30.81	37.98	32.55	37.40	33.54	33.21	34.64	33.46	34.11	32.36	33.16	34.83	32.56	32.32
	Ours, w/o L_p	33.32	30.50	37.97	32.42	37.41	33.21	33.10	34.76	33.16	33.73	32.29	33.14	34.86	32.61	32.32
SSIM ↑	AN	0.963	0.969	0.971	0.977	0.965	0.966	0.984	0.983	0.978	0.979	0.979	0.964	0.980	0.972	0.975
	UV	0.977	0.979	0.987	0.977	0.988	0.976	0.981	0.985	0.977	0.984	0.982	0.977	0.984	0.981	0.980
	Ours, w/ L_p	0.990	0.989	0.992	0.989	0.994	0.984	0.992	0.993	0.992	0.994	0.991	0.989	0.993	0.990	0.989
	Ours, w/o L_p	0.990	0.989	0.993	0.988	0.994	0.983	0.992	0.994	0.991	0.994	0.992	0.990	0.993	0.989	0.989
LPIPS ↓	AN	0.041	0.034	0.036	0.040	0.027	0.061	0.026	0.023	0.029	0.024	0.026	0.028	0.042	0.056	0.053
	UV	0.029	0.021	0.018	0.027	0.016	0.032	0.025	0.021	0.026	0.021	0.027	0.028	0.021	0.022	0.020
	Ours, w/ L_p	0.015	0.013	0.012	0.021	0.011	0.030	0.015	0.017	0.019	0.011	0.014	0.015	0.016	0.018	0.014
	Ours, w/o L_p	0.036	0.030	0.026	0.041	0.021	0.054	0.024	0.021	0.027	0.020	0.024	0.027	0.032	0.038	0.036

result in substantial errors. To this end, we impose the blend weight loss to suppress the standard deviation of blend weights in each Gaussian. Specifically, for each Gaussian G_i, $\mathbf{s}$ corresponds to three scales $\{s_1, s_2, s_3\}$ along the Gaussian axes $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$. we fetch six points along the three axes: $\mathbf{p}_i^{\pm l \in 1,2,3} = \mathbf{x_i} \pm s_l \mathbf{a}_l$, and compute their corresponding blend weights $\{\mathbf{w}(\mathbf{p}_i^{\pm 1}), \mathbf{w}(\mathbf{p}_i^{\pm 2}), \mathbf{w}(\mathbf{p}_i^{\pm 3})\}$. The blend weight loss is defined as the standard deviations of the six weights on all Gaussians:

$$L_w = \sum_{i=i}^{N} \mathrm{std}(\mathbf{w}(\mathbf{p}_\mathrm{i}^{\pm 1}), \mathbf{w}(\mathbf{p}_\mathrm{i}^{\pm 2}), \mathbf{w}(\mathbf{p}_\mathrm{i}^{\pm 3})). \tag{8}$$

The D-SSIM term L_{D-SSIM} is the same as that in 3DGS [Kerbl et al. 2023]. L_p is the perceptual loss [Zhang et al. 2018] as in UV Volume [Chen et al. 2023], which is optional and would generate results with better human visual perception but at the cost of more training time. We compare the results generated with and without the perceptual loss in Sec. 4.

3.4 Implementation Details

We adopt shallow 4-layer MLPs with ReLU activations for both F_a and $F_{\Delta w}$. The hidden layer width of the MLP F_a and $F_{\Delta w}$ is 128 and 32, respectively. The dimension of the latent code $\mathbf{f}$ is 9, which is initialized by positional encoding [Mildenhall et al. 2020] using the position of the Gaussian center at the beginning.

To enhance the training stability, we divide the training process into two stages. In the first stage (5000 iterations), we disable the Gaussian correction MLP F_a and residual weight prediction MLP $F_{\Delta w}$, and only optimize the basic Gaussian properties, as well as the joint parameters in input video frames, which is equivalent to computing an average human model across all input video frames. In the second stage, we enable the two MLPs and jointly optimize the MLP, Gaussian and joint parameters. It should also be noted that in (1), we stop the gradient of Θ to disentangle poses from the appearance information.

To avoid falling into local optima, we reset Gaussian opacities every fixed number of iterations, similar to 3DGS [Kerbl et al. 2023]. In the first stage, we reset the opacity at the 3000th iteration. In the second stage, we reset the opacity every 6000 iterations. We apply a similar Gaussian densification and pruning strategy as in 3DGS. The difference is that we use the sum of the gradients from the alpha and RGB loss to determine whether to densify Gaussians, which accelerates the fitting of deformable garments. In addition, we use the basic Gaussian scales and opacities (i.e., without the corrected values from F_a) to split or prune Gaussians, which keeps the consistency across all poses.

4 EXPERIMENTS

We conduct experiments on a workstation with an i7-13700KF CPU, 32GB memory and an NVIDIA RTX 4090 GPU, to demonstrate the effectiveness and efficiency of our method. We present quantitative results measured with three standard metrics: PSNR, SSIM and LPIPS. Note that we use the whole image including the black background region instead of the masked image for metric evaluation.

Dataset. We perform experiments on the ZJU Mocap dataset [Peng et al. 2021b], H36M dataset [Ionescu et al. 2013], and CMU Panoptic dataset [Joo et al. 2017], which include multi-view sequences, calibrated camera parameters, masks and poses (estimated by Easy-Mocap [eas 2021]). We use 20 training views on the ZJU Mocap dataset with 512×512 resolution and the CMU Panoptic dataset with 1920 × 1080 resolution. To test our method under sparse view input, we only use 3 views for training on H36M dataset with 1000 × 1000 resolution.

Baselines. We validate our method by comparing it with two representative human avatar synthesis methods: 1) AN: Animatable NeRF [Peng et al. 2021a]; 2) UV: UV Volume [Chen et al. 2023].

Table 2. Quantitative results of novel pose synthesis. Our method outperforms baselines (AN [Peng et al. 2021a] and UV [Chen et al. 2023]) on PSNR and SSIM (especially on H36M dataset) and presents competitive LPIPS. After adding the perceptual loss, our method shows the best LPIPS while the PSNR and SSIM are worse than our method without the perceptual loss.

Datasets		ZJU						H36M						CMU		
		313	315	386	387	390	392	s1p	s5p	s6p	s7p	s8p	s9p	p1	p4	p6
PSNR ↑	AN	30.81	28.24	35.20	29.55	33.88	31.32	31.34	32.66	32.15	30.24	30.88	31.47	28.83	27.66	27.54
	UV	30.75	28.30	34.92	29.62	34.44	31.51	30.38	31.84	30.96	30.27	30.95	30.92	29.64	27.98	28.68
	Ours, w/ L_p	32.67	30.01	36.03	31.15	34.92	32.68	37.12	35.88	35.85	36.04	36.42	36.96	30.54	28.89	29.68
	Ours, w/o L_p	33.41	30.17	36.85	31.76	35.57	32.76	37.86	36.77	35.86	36.63	36.48	37.39	31.66	29.61	30.22
SSIM ↑	AN	0.971	0.973	0.969	0.975	0.970	0.970	0.990	0.988	0.986	0.983	0.985	0.991	0.953	0.958	0.941
	UV	0.968	0.974	0.984	0.972	0.985	0.971	0.988	0.988	0.984	0.987	0.987	0.987	0.969	0.963	0.964
	Ours, w/ L_p	0.987	0.987	0.990	0.985	0.992	0.981	0.990	0.989	0.995	0.987	0.997	0.994	0.982	0.979	0.980
	Ours, w/o L_p	0.988	0.988	0.992	0.987	0.993	0.983	0.997	0.996	0.995	0.996	0.997	0.996	0.988	0.983	0.987
LPIPS ↓	AN	0.042	0.033	0.032	0.044	0.028	0.062	0.022	0.020	0.019	0.021	0.016	0.018	0.068	0.078	0.071
	UV	0.039	0.024	0.020	0.030	0.017	0.039	0.019	0.017	0.018	0.017	0.017	0.018	0.039	0.049	0.038
	Ours, w/ L_p	0.017	0.010	0.012	0.021	0.014	0.036	0.009	0.009	0.011	0.008	0.011	0.008	0.039	0.042	0.039
	Ours, w/o L_p	0.036	0.030	0.030	0.045	0.026	0.057	0.013	0.017	0.018	0.014	0.013	0.015	0.053	0.062	0.057

Table 3. Comparison of the training and rendering performance of baselines (AN [Peng et al. 2021a] and UV [Chen et al. 2023]) and our method.

	AN	UV	Ours (w/o L_p)	Ours (w/ L_p)
Training Time	18 h	19 h	1.2 h	1.6 h
Novel View Syn.	3 fps	52 fps	114 fps	110 fps
Novel Pose Syn.	0.06 fps	18 fps	66 fps	60 fps

4.1 Comparisons with Baselines

Efficiency. Table 3 compares the training and rendering performance of baselines with our method on the ZJU Mocap dataset. As shown, our method achieves the highest FPS, enabling real-time rendering on both novel view and novel pose synthesis tasks. For novel view synthesis, we can cache the outputs of the Gaussian correction MLP F_a and reach the same FPS reported in 3DGS [Kerbl et al. 2023]. The training of our method converges within 100 minutes, while the baselines need more than 10 hours. Note that without the perceptual loss, our training time will reduce by approximately 30 minutes.

Novel View Synthesis. We synthesize novel views of training video frames. As shown in Table 1, our method is consistently superior to baselines in terms of all metrics. Particularly, in the case of sparse training views of the H36M dataset, our method outperforms [Chen et al. 2023; Peng et al. 2021a] approximately by a margin of 4 in terms of the PSNR metric, clearly demonstrating the generalization ability of our method. Omitting the perceptual loss in our training does not cause noticeable affection to the PSNR and SSIM metrics, but leads to significant increases in the LPIPS metric. As UV Volume uses the perceptual loss, it achieves better results in terms of the LPIPS metric than our method without the perceptual loss training.

Fig. 5 presents the qualitative comparison of our method with baselines. In all examples, Animatable NeRF [Peng et al. 2021a] generates blurry results and some parts of the body can even disappear. UV Volume is better at synthesizing texture and wrinkle details, but may introduce details unseen in ground truth images (see Fig. 1). More importantly, we observe that the texture prediction of UV Volume is inconsistent across different poses, resulting in jittering artifacts in synthesized videos (see the supplementary video). Our method is able to synthesize videos of better visual quality without temporal jittering.

Novel Pose Synthesis. We synthesize images with novel poses unused in training video frames. The quantitative results are shown in Table 2. Similar to novel view synthesis, our method performs the best in novel pose synthesis, in terms of all metrics. The PSNR improvements can be up to 6.5 on the H36M dataset, clearly demonstrating the superiority of our method over baselines.

The qualitative comparisons are shown in Fig. 4. Animatable NeRFF [Peng et al. 2021a] fails to preserve high-frequency details. UV Volume exhibits apparent artifacts in novel pose synthesis, such as arms of varying widths. In contrast, our method preserves detailed spatially-varying textures of clothes and always demonstrates robust shapes of body and limbs.

4.2 Ablation Studies

We conduct ablation studies on sequence 313 of the ZJU Mocap dataset. We validate the impacts of several possible choices, including simple combination of 3DGS and LBS, using positional encoding as the input of the Gaussian correction MLP F_a instead of the latent code, only computing image loss in the masked region, and correcting the higher-order components of spherical harmonics with F_a. We also validate the necessity of some modules in our method, including the optimization of joint parameters, Gaussian correction model F_a, boundary mask, alpha loss and blend weight loss, respectively. The quantitative results are summarized in Table 4 and the qualitative results are illustrated in Fig. 6.

Simple combination of 3DGS and LBS. We test a simple combination of 3DGS and LBS by omitting the Gaussian correction model

Table 4. Quantitative results of ablation studies.

Ablations	Novel View Synthesis			Novel Pose Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Complete Model	33.32	0.989	0.036	33.41	0.987	0.036
Simple 3DGS+LBS	28.56	0.976	0.083	29.56	0.976	0.081
Positional Encoding	30.03	0.981	0.053	30.32	0.978	0.053
Only Foreground	24.22	0.291	0.163	24.26	0.280	0.160
All-order SH	33.31	0.989	0.038	32.73	0.986	0.039
w/o First Stage	25.80	0.965	0.077	25.81	0.961	0.080
w/o Joints Optim.	32.42	0.986	0.053	32.19	0.984	0.051
w/o Boundary Mask	32.10	0.986	0.044	32.41	0.984	0.044
w/o F_a	29.50	0.981	0.075	29.99	0.979	0.073
w/o L_α	33.21	0.988	0.040	32.01	0.985	0.042
w/o L_w	33.32	0.989	0.041	33.04	0.986	0.043

and human joint optimization. As shown in Fig. 6, this approach cannot model fine details very well and exhibits joint dislocation in synthesized images.

Positional encoding versus Latent code. Our Gaussian correction MLP F_a is designed to compute the property changes of each Gaussian under a target pose. A simple approach is to use positional encoding [Mildenhall et al. 2020] of each Gaussian position and the target pose as the input to F_a. Compared with learning a latent code for each Gaussian, positional encoding generates inferior results lack of fine details. Using positional encoding of higher dimensions (21 in our experiment) or deeper MLP may alleviate this problem but would severely reduce the real-time performance.

Image loss from the masked region only versus alpha loss L_α. Our alpha loss L_α is designed to explicitly prevent Gaussians from growing out of the human region. Computing the image loss from the masked region without using L_α would produce Gaussians locating outside of the masked region, which do not affect the image loss, but still contribute to the final rendering and cause artifacts in novel view or novel pose synthesis.

Correcting all-order versus zero-order components of spherical harmonics. To disentangle pose-dependent and view-dependent appearance variations, we only correct the zero-order component related to pose transformation by F_a. As illustrated in Fig. 6, correcting all-order components leads to noisy results.

Necessity of human joint optimization. The joint parameters provided by the dataset are estimated by EasyMocap [eas 2021], which may not be very accurate. Inaccurate joint positions could cause unmooth bending of joints and inaccurate joint rotations lead to blurring.

Necessity of the boundary mask. Gaussian splatting naturally has a smooth color fade, while the binary human masks have aliasing and mutation on the boundary. If we force Gaussians to directly fit the masked image, lots of tiny Gaussians will be generated to match the boundary, which do not improve the image quality, but bring noises.

Impacts of F_a and losses. Without the Gaussian correction model F_a, we can only model the average human appearance across all training video frames and lose all pose-dependent appearance details. The alpha loss not only constrains the position of Gaussians and removes the background interference, but also makes our method able to better model garment movements. The blend weight loss ensures that each Gaussian can undergo a transformation as a cohesive unit, which prevents Gaussians from protruding outside the body in novel poses.

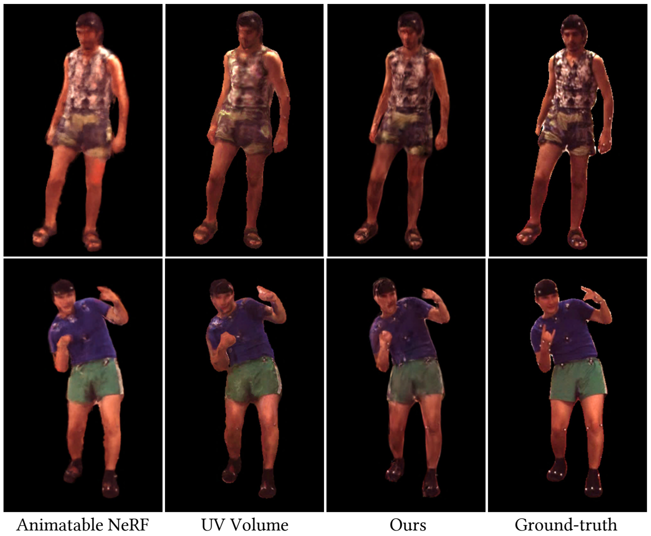

Fig. 3. Qualitative results of novel view synthesis on the H36M dataset. Due to the lack of sufficient training views, the synthesized images are blurry and distorted in test views.

4.3 Limitation

Our method is able to synthesize high quality and temporally stable human avatars in both novel views and novel poses. However, when the training views are too sparse, our method tends to overfit and generates inferior results in novel views. Nevertheless, as shown in Fig. 3, our method still synthesizes stable body shapes and clothing details, outperforming baselines in terms of all metrics. Besides, when the garment wrinkles change rapidly, our method may produce a transition with observable noises. Adding more constraints or introducing prior information may alleviate these problems.

5 CONCLUSION

We present an animatable 3D Gaussian representation for rendering high-quality free-view dynamic humans in real time. It can well synthesize high-frequency and pose-consistent human appearance details. Each Gaussian within the representation is associated with a few basic properties representing the average human appearance, a latent code for Gaussian correction to reflect appearance changes under novel poses, and a set of blend weights for transforming Gaussians to target poses with LBS. Experiments on three popular datasets demonstrate that our model achieves the best image quality and rendering performance in novel view synthesis of dynamic humans under novel poses.

REFERENCES

2021. EasyMoCap - Make human motion capture easier. Github. https://github.com/zju3dv/EasyMocap

Brett Allen, Brian Curless, and Zoran Popović. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)* 22, 3 (2003), 587–594.

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

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2023. Unstructured lumigraph rendering. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 497–504.

Yue Chen, Xuan Wang, Xingyu Chen, Qi Zhang, Xiaoyu Li, Yu Guo, Jue Wang, and Fei Wang. 2023. UV Volumes for real-time rendering of editable free-view human performance. In *Proceedings of CVPR*. 16621–16631.

Abe Davis, Marc Levoy, and Fredo Durand. 2012. Unstructured light fields. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 305–314.

Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. 2008. Floating textures. In *Computer graphics forum*, Vol. 27. Wiley Online Library, 409–418.

Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14346–14355.

Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. 2012. Drape: Dressing any person. *ACM Transactions on Graphics (ToG)* 31, 4 (2012), 1–10.

Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2021. Real-time deep dynamic characters. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–16.

Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. 2023. GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians. *arXiv preprint arXiv:2312.02134* (2023).

Shoukang Hu and Ziwei Liu. 2023. GauHuman: Articulated Gaussian Splatting from Monocular Human Videos. *arXiv preprint arXiv:2312.02973* (2023).

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* 36, 7 (2013), 1325–1339.

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and John P Lewis. 2014. Skinning: Real-time shape deformation (full text not available). In *ACM SIGGRAPH 2014 Courses*. 1–1.

Rohit Jena, Ganesh Subramanian Iyer, Siddharth Choudhary, Brandon Smith, Pratik Chaudhari, and James Gee. 2023. SplatArmor: Articulated Gaussian splatting for animatable humans from monocular RGB videos. *arXiv preprint arXiv:2311.10812* (2023).

Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. 2023. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16922–16932.

Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2017. Panoptic Studio: A Massively Multiview System for Social Interaction Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

Hanbyul Joo, Tomas Simon, and Yaser Sheikh. 2018. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8320–8329.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)* 42, 4 (2023), 1–14.

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

Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. 2023. Gart: Gaussian articulated template models. *arXiv preprint arXiv:2311.16099* (2023).

Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. 2023. Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling. *arXiv preprint arXiv:2311.16096* (2023).

Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2022. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2021. Real-time high-resolution background matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8762–8771.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751* (2019).

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2023. SMPL: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 851–866.

Shugao Ma, Tomas Simon, Jason Saragih, Dawei Wang, Yuecheng Li, Fernando De La Torre, and Yaser Sheikh. 2021. Pixel codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 64–73.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. 2023. Human Gaussian Splatting: Real-time Rendering of Animatable Avatars. *arXiv preprint arXiv:2311.17113* (2023).

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15.

Ahmed AA Osman, Timo Bolkart, and Michael J Black. 2020. Star: Sparse trained articulated human body regressor. In *Computer Vision–ECCV 2020: 16th European Conference, 2020, Proceedings, Part VI 16*. Springer, 598–613.

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of ICCV*. 14314–14323.

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9054–9063.

Ken Shoemake and Tom Duff. 1992. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface*, Vol. 92. 258–264.

Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. 2022. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13524–13534.

Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. 2023. StyleAvatar: Real-time Photo-realistic Portrait Avatar from a Single Video. *arXiv preprint arXiv:2305.00942* (2023).

Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. 2020. Multi-view neural human rendering. In *Proceedings of CVPR*. 1682–1691.

Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. 2023. FlashAvatar: High-Fidelity Digital Avatar Rendering at 300FPS. *arXiv preprint arXiv:2312.02214* (2023).

Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. 2011. Video-based characters: creating new human performances from a multi-view video database. In *ACM SIGGRAPH 2011 papers*. 1–10.

Zhen Xu, Sida Peng, Chen Geng, Linzhan Mou, Zihan Yan, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. 2023a. Relightable and Animatable Neural Avatar from Sparse-View Video. *arXiv preprint arXiv:2308.07903* (2023).

Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 2023b. 4K4D: Real-Time 4D View Synthesis at 4K Resolution. *arXiv preprint arXiv:2310.11448* (2023).

Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. 2022. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2863–2873.

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

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5752–5761.

Chao Zhang, Sergi Pujades, Michael J Black, and Gerard Pons-Moll. 2017. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4191–4200.

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

Fuqiang Zhao, Wei Yang, Jiakai Zhang, Pei Lin, Yingliang Zhang, Jingyi Yu, and Lan Xu. 2022. Humannerf: Efficiently generated human radiance field from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7743–7753.

Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. 2022. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15893–15903.

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

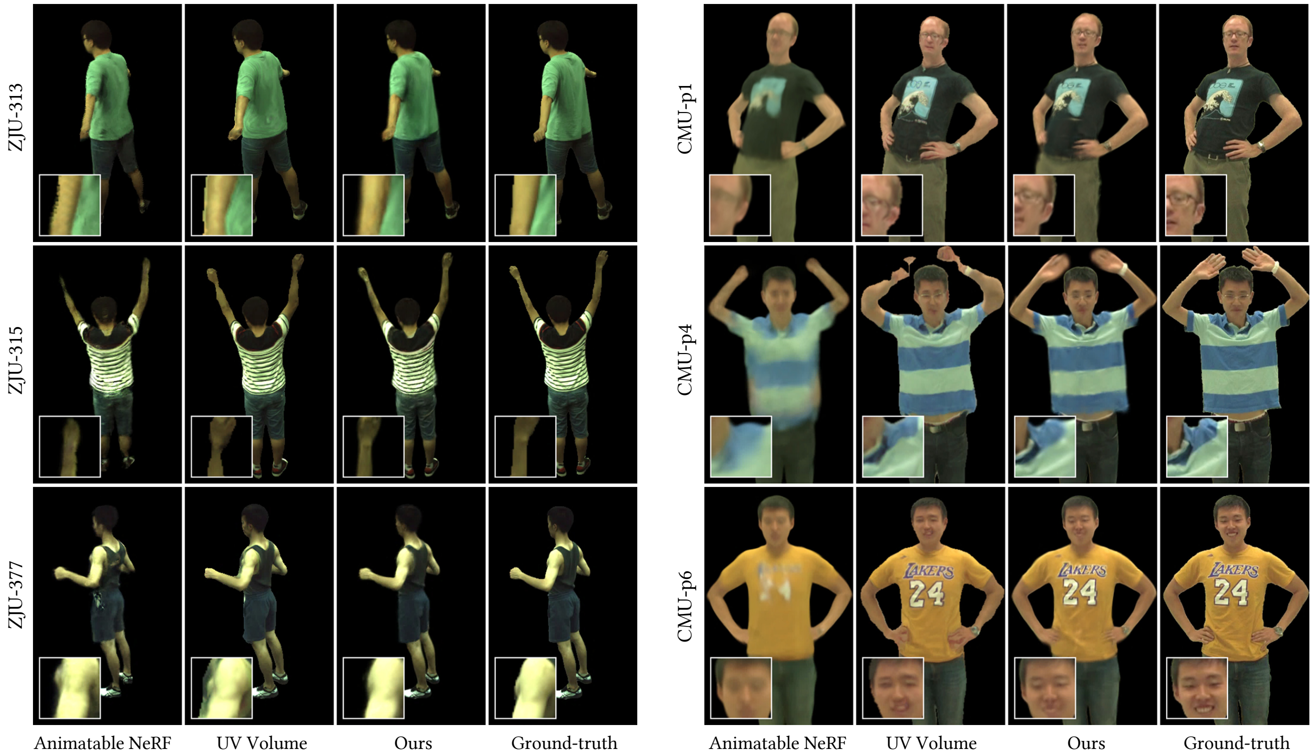

Fig. 4. Qualitative results of novel pose synthesis on the ZJU Mocap and CMU Panoptic datasets.

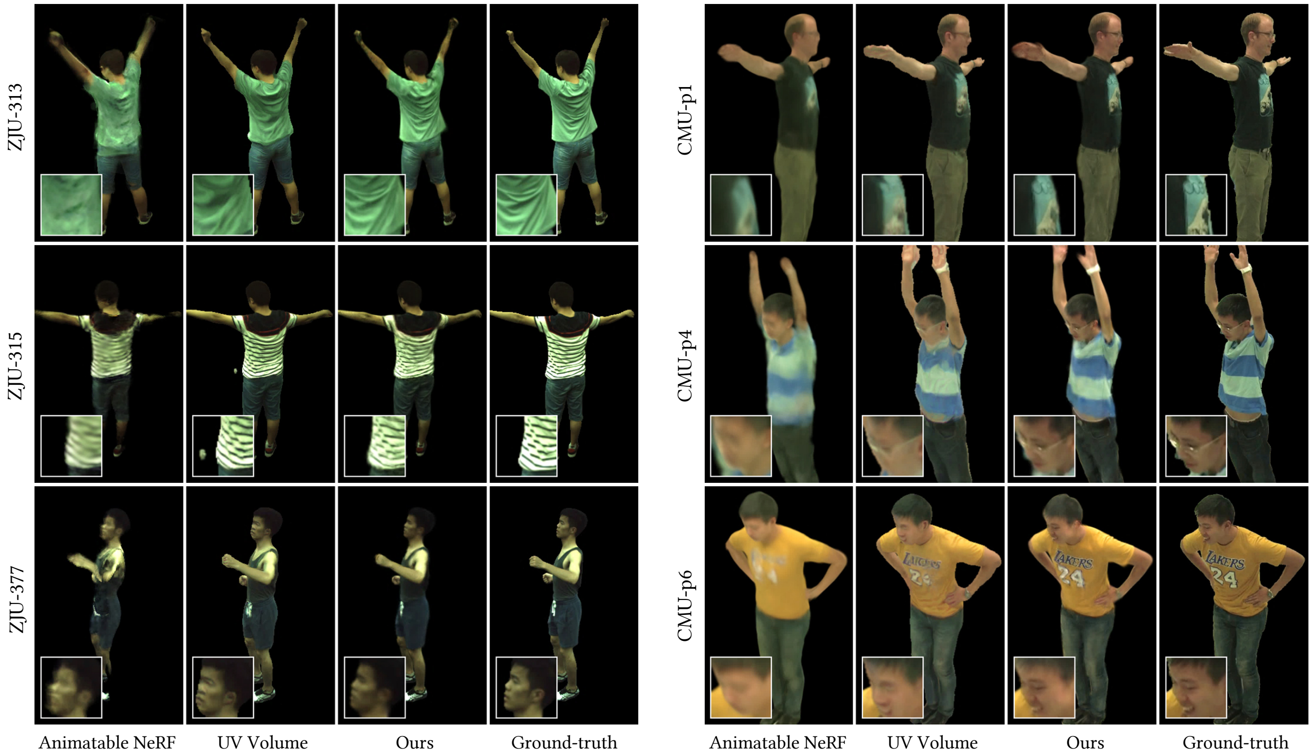

Fig. 5. Qualitative results of novel view synthesis on the ZJU Mocap and CMU Panoptic datasets.

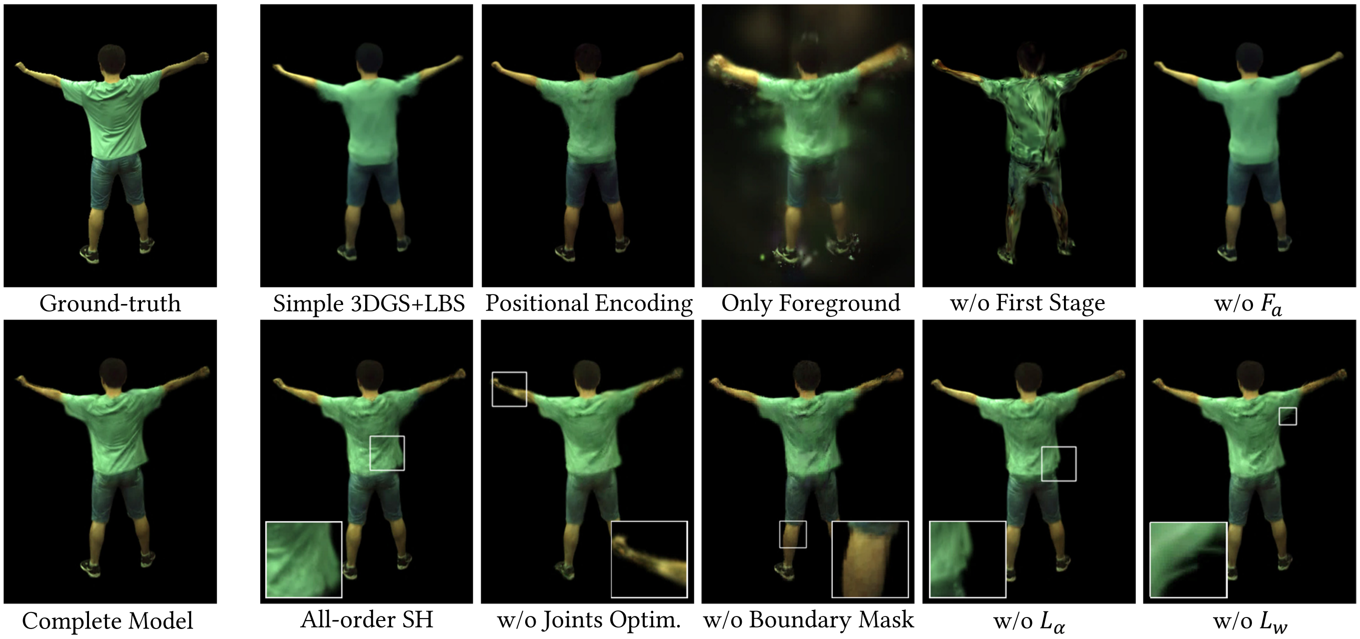

Fig. 6. Qualitative results of ablation studies. Simple combination of 3DGS and LBS (Simple 3DGS+LBS) produces blurring and joint dislocation. Using positional encoding instead of the latent code or not using F_a also leads to blurring. Only computing image loss from the masked foreground (Only Foreground) causes Gaussian to spread into empty space. Correcting all-order SH by F_a (All-order SH) decreases the efficiency of training, and also causes noises and artifacts in novel views. Without the initialization stage (w/o First Stage), training totally fails. Without joints and poses optimization, hands may disappear in some cases. Without the boundary mask, noises and zigzags appear on the boundary. Without the alpha loss L_α, the result does not capture the garment movement well. Without the blend weight loss L_w, some Gaussians protrude outside the body in the novel pose.