Real-time High-fidelity Gaussian Human Avatars with Position-based Interpolation of Spatially Distributed MLPs

Youyi Zhan[1] Tianjia Shao[1*] Yin Yang[2] Kun Zhou[1]
[1]State Key Lab of CAD&CG, Zhejiang University [2]University of Utah

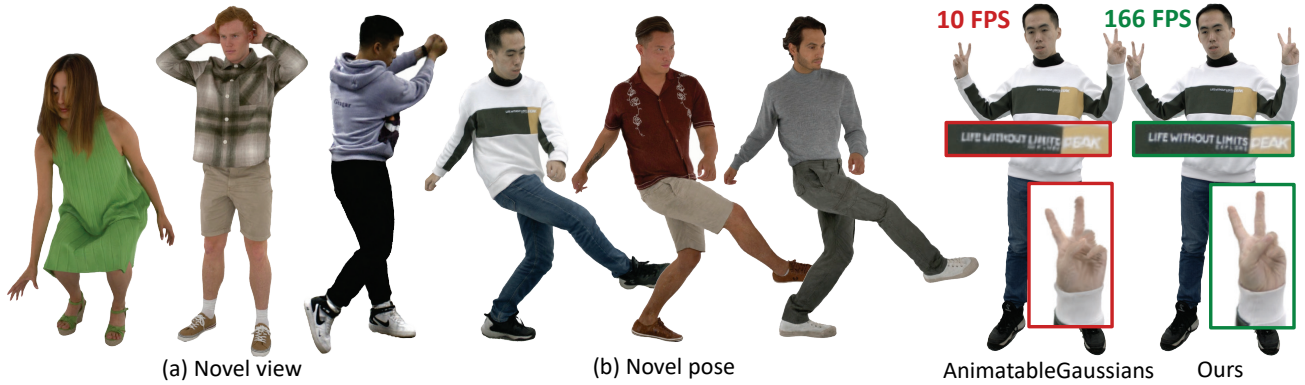

Figure 1. Our method can model high-fidelity human avatars that can be animated under novel poses and rendered in real-time. Compared to the state-of-the-art method AnimatableGaussians [28], our approach can recover finer details while achieving significantly faster rendering speed (166 fps) under novel views and novel poses.

Abstract

Many works have succeeded in reconstructing Gaussian human avatars from multi-view videos. However, they either struggle to capture pose-dependent appearance details with a single MLP, or rely on a computationally intensive neural network to reconstruct high-fidelity appearance but with rendering performance degraded to non-real-time. We propose a novel Gaussian human avatar representation that can reconstruct high-fidelity pose-dependence appearance with details and meanwhile can be rendered in real time. Our Gaussian avatar is empowered by spatially distributed MLPs which are explicitly located on different positions on human body. The parameters stored in each Gaussian are obtained by interpolating from the outputs of its nearby MLPs based on their distances. To avoid undesired smooth Gaussian property changing during interpolation, for each Gaussian we define a set of Gaussian offset basis, and a linear combination of basis represents the Gaussian property offsets relative to the neutral properties. Then we propose to let the MLPs output a set of coefficients corresponding to the basis. In this way, although Gaussian coefficients are derived from interpolation and change smoothly, the Gaussian offset basis is learned freely without constraints. The smoothly varying coefficients combined with freely learned basis can still produce distinctly different Gaussian property offsets, allowing the ability to learn high-frequency spatial signals. We further use control points to constrain the Gaussians distributed on a surface layer rather than allowing them to be irregularly distributed inside the body, to help the human avatar generalize better when animated under novel poses. Compared to the state-of-the-art method, our method achieves better appearance quality with finer details while the rendering speed is significantly faster under novel views and novel poses.

1. Introduction

Digital human avatars have widespread applications in the fields like virtual reality and visual content creation. The reconstruction of human avatars from multi-view videos has been extensively studied, and remarkable progress is being made with the radiance field representation (i.e., NeRF [33] and 3D Gaussians [20]). Specially, with the fast training and rendering speed, many works have succeeded in using 3D Gaussians to reconstruct digital human avatars with high quality pose-dependent appearances.

Among those works, a common way is to use a single multilayer perceptron (MLP) network [35, 42, 54] , which takes human pose along with Gaussian position or learnable code as input and outputs Gaussian property offsets for each Gaussian. Though these methods can achieve real-time ren-

*Corresponding author (tjshao@zju.edu.cn)

dering performance (i.e., 30-60 fps), due to limited learning capacity, they fail to capture high-frequency details. To reconstruct high-fidelity detailed appearances, the state-of-the-art work AnimatableGaussians [28] proposes to use more powerful network StyleUNet [47] to predict Gaussian property maps with the posed position map as input. AnimatableGaussians [28] can reconstruct the highest-quality human avatar among existing methods. However, due to heavy computational burden brought by StyleUNet, the rendering performance is not real time (i.e., around 10 fps).

Our goal is to reconstruct high-fidelity detailed Gaussian human avatars, which can be rendered in real time under novel views and novel poses. To this end, we propose a new Gaussian human avatar representation, empowered by spatially distributed MLPs. Different from the single MLP which takes the position (or learnable code) and pose as input, the spatially distributed MLPs are explicitly located on different anchor positions on human body, and their input is only the human pose. The parameters stored in each Gaussian are obtained by interpolating from the outputs of its nearby MLPs based on distances. In this way, each MLP is only responsible for learning the the human appearance of its local region, reducing the learning burden and enhancing the capability of capturing high-frequency details. Besides, with the position-based interpolation, we don't need to send every Gaussian position to the MLP anymore, avoiding going through the MLP too many times. For a pose, each MLP only needs to be computed once, so the rendering performance can be significantly accelerated, reaching 166 fps even with 200K Gaussians under novel poses.

However, what to interpolate is not trivial. If we let the MLPs output Gaussian property offsets as the single MLP in previous methods, and perform interpolation on the Gaussian property offsets, we will obtain smoothly changing Gaussian property offsets across the body, which is undesirable and will produce artifacts (see Fig. 6 for example). To this end, for each Gaussian, we define the neutral properties representing the mean appearance, and a set of Gaussian offset basis. A linear combination of the basis represents the Gaussian property offsets relative to the neutral properties. Then we propose to let the MLPs output a set of coefficients, and the Gaussian coefficients for the offset basis are obtained by interpolation. The key insight of our design is, although Gaussian coefficients are derived from interpolation and change smoothly, the Gaussian offset basis is learned freely without constraints. Therefore, the smoothly varying coefficients combined with freely learned basis can still produce distinctly different Gaussian property offsets. This allows the Gaussians to have the ability to learn high-frequency spatial signals.

Furthermore, we propose to use control points to constrain the Gaussians distributed on a surface layer rather than allowing them to be irregularly distributed inside the body. Specifically, the Gaussian position offset from the neutral position is not optimized freely, but is interpolated from the offsets of sampled control points on the body. By constraining the neighboring control points to have similar position offsets, the Gaussians among these control points are also constrained to move in the same direction without undesired moving inside. This design helps the human avatar generalize better and eliminates artifacts when animated under novel poses.

Experiments demonstrate that our method can reconstruct high-fidelity appearance with high-frequency details of human avatars. Compared to the state-of-the-art method, our method achieves better appearance quality with finer details while the rendering speed is significantly faster under novel views and novel poses (166 fps versus 10 fps).

2. Related Work

Mesh based Human Avatar. Using mesh with texture is the most common approach to model human avatars. Through learning from video, many methods reconstruct geometry for individuals and apply textures to obtain appearance. [1, 5, 51, 52] use dense camera arrays to reconstruct geometry. [2, 46] further use depth cameras to assist in geometry reconstruction. To obtain appearances under different poses, [10, 11, 31, 51–53] use neural networks to output textures for different poses. [51–53] further model clothing as a separate layer, achieving realistic results with garment dynamics.

Neural Rendering for Human Avatar. In recent years, neural radiance field (NeRF [33]) has been widely used for human avatar reconstruction. Many methods [18, 25, 27, 29, 30, 40, 41, 50, 55, 58] render the human avatar by inverse LBS and obtaining attributes (like color and density) for volume rendering, and achieve good results. ARAH [48] and Vid2avatar [9] further use signed distance function (SDF) to represent human geometry. However, due to the requirement of multiple sampling, these methods are inefficient, resulting in several seconds to render a single image.

Some works focus on accelerating the above rendering process to achieve faster training and rendering speeds. InstantNVR [8] and InstantAvatar [17] use iNGP network [36] to speed up training. However, these methods fail to capture human details, resulting in less realistic rendering. AvatarRex [59], Deliffas [22], UV Volume [3] and RAM-Avatar [6] achieve high-quality appearances at real-time speeds. AvatarRex [59] is a method for full body avatar with face, body and hands, where SLRF [58] and dynamic feature patches are used to model the body geometry and color. Despite some breakthroughs in speed, these methods still rely on extensive network computation to obtain appearances, limiting their speed to around 10-25 fps.

3DGS based Human Avatar. 3D Gaussian Splatting (3DGS [20]) provides a new paradigm for scene reconstruc-

tion. Many works have successfully use 3DGS for modeling human avatars. Kwon et al. [23] and GPS-Gaussian [57] are able to reconstruct the avatar from multi-view cameras, but their avatar cannot be driven by novel poses. SplattingAvatar [44], HAHA [45], GomAvatar [49], Moon et al. [34], EVA [13], Gauhuman [14], GART [24], iHuman [38], HUGS [21] and SplatArmor [16] propose to reconstruct human avatars from monocular video. However, they cannot model pose-dependent appearance. 3DGS-Avatar [42], Ye et al. [54], Moreau et al. [35] propose to use a single MLP to output Gaussian property offsets under different poses. They typically use pose along with positions [42] or per-Gaussian learnable codes [35, 54] as MLP inputs to predict Gaussian properties. However, these methods fail to capture high-frequency details due to limited learning capacity. Ash [37], UV Gaussians [19], MeshAvatar [4] proposes to use convolutional neural network (CNN) to output Gaussian property maps, but these works still fail to reconstruct realistic human avatars. Both AnimatableGaussians [28] and DEGAS [43] use large CNN to learn appearance and achieve high-quality rendering. However, it is slow to go through their network, as their large CNNs involve substantial computation, limiting the rendering speed. Instead, our approach doesn't involve heavy neural network computations and can render high-quality avatar at faster speed.

Discussion of Highly Related Works. For the MLPs, both SLRF [58] and our method employ spatially distributed MLPs to learn local appearance. However, SLRF takes position encoding as MLP input, which requires a huge amount of position queries to the MLPs, largely decreasing the inference speed, while our MLPs only take the human pose as input, significantly reducing the computational burden. For the linear basis, Gao et al. [7] and Ma et al. [32] learn the blendshape basis while the coefficients are from the FLAME [26] model and not learnable. Our method jointly learns the coefficients and basis.

3. Method

Our approach takes the multi-view videos of a person as input. Following previous methods [15, 28, 59], we extract the foreground human mask, and register the SMPL-X [39] model for each frame to obtain the 3D human pose. We also use the method of AnimatableGaussians [28] to obtain a canonical template mesh. Our goal is to reconstruct a human avatar, which has pose-dependent high-fidelity detailed appearances under novel views and novel poses, and meanwhile can be rendered in real-time. We first introduce our Gaussian avatar representation with spatially distributed MLPs (Sec. 3.1). Then we propose to use control points to obtain per Gaussian position offset, so that Gaussians can be constrained on a surface layer (Sec. 3.2). Finally, we describe the training and testing process, as well as implementation details (Sec. 3.3).

3.1. Gaussian Avatar with Spatially Distributed MLPs

Our avatar is composed of N Gaussians and F spatially distributed MLPs. Each Gaussian has a set of neutral properties Λ_0, including rotation $\mathbf{r}_0$, scale $\mathbf{s}_0$, opacity o_0, SH coefficients $\mathbf{c}_0$, and position $\mathbf{x}_0$. The neutral properties represent the mean human appearance across the video frames. We also define a set of Gaussian offset basis $\delta\Lambda^k = \{\delta\mathbf{r}^k, \delta\mathbf{s}^k, \delta o^k, \delta\mathbf{c}^k\},\ k \in [1, B]$ for each Gaussian. A linear combination of the basis represents a Gaussian property change relative to the neutral properties.

The spatially distributed MLPs are located on F anchor points $\{\mathbf{x}_a^j\}_{j\in[1,F]}$ uniformly sampled on the template mesh. Each MLP is only responsible for learning the local appearance change around the anchor point. The MLP takes the human pose vector $\boldsymbol{\theta}$ as input, and outputs the anchor coefficients $\mathbf{w}_a^j$ on each anchor point,

$$\mathbf{w}_a^j = \mathcal{E}^j(\boldsymbol{\theta}), \tag{1}$$

where $\mathcal{E}^j$ is the spatially distributed MLP located on the jth anchor point. Based on the anchor coefficients, we obtain the Gaussian coefficients $\mathbf{w}_g$ for the offset basis on each Gaussian by interpolating from the nearest three anchor points,

$$\mathbf{w}_g = \frac{\sum_j \gamma(\mathbf{x}_0, \mathbf{x}_a^j) \cdot \mathbf{w}_a^j}{\sum_j \gamma(\mathbf{x}_0, \mathbf{x}_a^j)}, \tag{2}$$

where $\gamma(\mathbf{x}, \mathbf{y}) = 1/\|\mathbf{x} - \mathbf{y}\|_2$ is the reciprocal of the distance between two points. j is the index of three nearest anchor points.

We linearly combine the Gaussian offset basis using the Gaussian coefficients to obtain the property offset under pose $\boldsymbol{\theta}$ for each Gaussian, and the Gaussian properties are obtained by adding the property offset to the neutral Gaussian,

$$\begin{aligned} \delta\Lambda &= \sum_{k=1}^{B} \mathbf{w}_g[k] \cdot \delta\Lambda^k \\ \Lambda &= \Lambda_0 + \delta\Lambda. \end{aligned} \tag{3}$$

Please note the Gaussian position is also computed as $\mathbf{x} = \mathbf{x}_0 + \delta\mathbf{x}$, nevertheless the position offset $\delta\mathbf{x}$ is actually interpolated from the position offsets of control points, which is detailed in Sec. 3.2.

Afterwards, the Gaussians are transformed from the canonical space to the pose $\boldsymbol{\theta}$ using linear blend skinning (LBS). The transformed Gaussians are finally rasterized to produce high-fidelity detailed human images.

3.2. Control Point

We do not calculate the position offset using Eq. (3) because this would allow each Gaussian to move freely in space during optimization. Irregularly distributed Gaussians in space

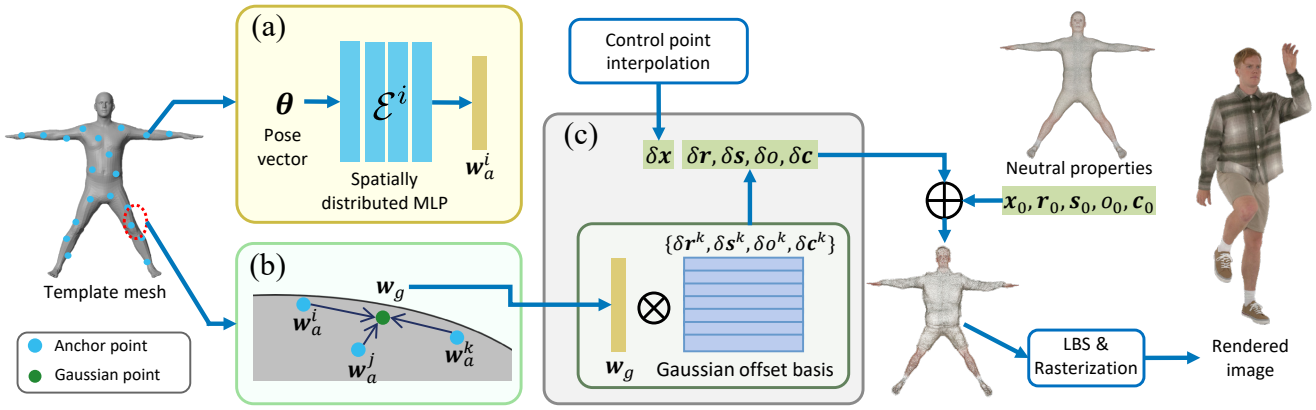

Figure 2. Pipeline overview. (a) We define the spatially distributed MLPs on anchor points, which are uniformly sampled on the template mesh. Each MLP takes the pose $\boldsymbol{\theta}$ as input and outputs the anchor coefficients $\mathbf{w}_a$. (b) The Gaussian coefficients $\mathbf{w}_g$ are interpolated from the coefficients of three nearest anchor points. (c) The Gaussian property offsets are obtained by linearly combining Gaussian offset basis using Gaussian coefficients. Then the neutral Gaussian properties are added with Gaussian property offsets to model the human appearance under pose $\boldsymbol{\theta}$. Finally the Gaussians are transformed to the pose $\boldsymbol{\theta}$ and rasterized to produce high-fidelity images. Note that the Gaussian position offset $\delta\mathbf{x}$ is obtained through control point interpolation, which is illustrated in Sec. 3.2.

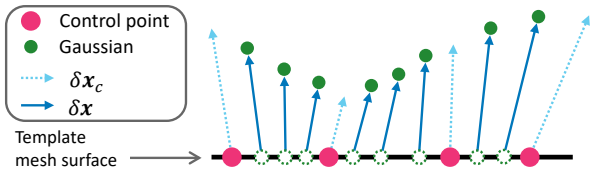

Figure 3. Illustration of the control point. The Gaussian position offset $\delta\mathbf{x}$ is interpolated from the position offsets of nearby control points $\delta\mathbf{x}_c$.

can produce non-neglectable artifacts (see Fig. 7 "w/o control point"). Therefore, we need to constrain the Gaussians distributed on a surface layer rather than allowing them to be irregularly distributed inside the body.

A straightforward solution is to use smoothness loss to constrain neighboring Gaussians to have similar position offsets from the neutral positions. However, we find such design can only ensure very local position smoothness and cannot prevent Gaussians from moving inside the body, resulting in suboptimal results (see Fig. 7 "w/o control point (w/ smooth)" and supplementary video). To this end, we propose to use sampled control points to yield similar position offsets across larger areas, and interpolate the position offsets on control points to generate the Gaussian position offsets.

Specifically, we uniformly sample C control points $\{\mathbf{x}_c^i\}_{i\in[1,C]}$ on the template mesh. Each control point has its neutral position offset $\delta\mathbf{x}_{c0}$, as well as a set of position offset basis $\{\delta\mathbf{x}_{cb}^k\}_{k\in[1,B]}$. We compute the position offset $\delta\mathbf{x}_c$ for each control point by using the control point coefficients $\mathbf{w}_c$ to combine the offset basis and adding the neutral offset,

$$\delta\mathbf{x}_c = \delta\mathbf{x}_{c0} + \sum_{k=1}^{B} \mathbf{w}_c[k] \cdot \delta\mathbf{x}_{cb}^k. \tag{4}$$

$\mathbf{w}_c$ is computed by interpolating the anchor coefficients of position offsets, similar to Eq. (2). Note the anchor coefficients of position offsets are simultaneously outputted from the position-ware MLP $\mathcal{E}^j(\boldsymbol{\theta})$ in Eq. (1)

Then, the Gaussian position offset $\delta\mathbf{x}$ is obtained by interpolating the position offsets of its three nearest control points,

$$\delta\mathbf{x} = \frac{\sum_i \gamma(\mathbf{x}_0, \mathbf{x}_c^i) \cdot \delta\mathbf{x}_c^i}{\sum_i \gamma(\mathbf{x}_0, \mathbf{x}_c^i)}, \tag{5}$$

where i is the index of the three nearest control points.

Since the position offset of a Gaussian is interpolated from nearby control points, by constraining the neighboring control points to have similar position offsets, the Gaussians among these control points are also constrained to move in the same direction. This ensures that the Gaussians can be optimized being distributed on a surface layer. Fig. 3 provides an illustration of the control point design.

Discussion. Note an alternative approach is to utilize another MLP to predict the position offsets for each control point. However, this approach produces blurry results (see Fig. 7 "w/ MLP position offset"). This is because using an MLP to learn position offsets for all control points in a sequence can exceed the network's learning capacity, making it struggle to learn position offsets for each control point.

3.3. Training and Testing

Implementation Details. We initialize $N = 200K$ Gaussians on the template mesh, with each Gaussian assigned the skinning weights according to AnimatableGaussians [28]. We also uniformly sample $F = 300$ anchor points and $C = 10K$ control points on the template mesh. The Gaussian neutral positions $\mathbf{x}_0$, anchor points $\mathbf{x}_a$, and control points $\mathbf{x}_c$ are fixed and not optimized once sampled on

the mesh. The spatially distributed MLP has four layers. The pose vector used as MLP input does not include finger joints, as we believe that changes in fingers do not affect the overall appearance. The basis number B is set to 15. Each sequence is trained for 800K iterations. Our method is implemented using PyTorch, and we are able to achieve fast rendering speed without using acceleration techniques like CUDA or TensorRT, unlike previous method [59].

Training. During training, we simultaneously learn the spatially distributed MLPs, Gaussian neutral properties and property offset basis, as well as the neural position offsets and position offset basis for control points. We set the learning rate as 5×10^{-4} for $\{\mathbf{r}_0, \mathbf{s}_0, o_0, \mathbf{c}_0\}$, 1.6×10^{-4} for $\delta\mathbf{x}_{c0}$, and 5×10^{-4} for spatially distributed MLPs. The learning rates for the basis are five times smaller. We also use learning rate decay in our implementation.

For the loss functions, we use the L1 loss as in 3DGS [20] and LPIPS [56] loss. The nearby control points are constrained to have similar position offsets $\delta\mathbf{x}_c$:

$$\mathcal{L}_{ctrl} = \sum_{i,j} \|\delta\mathbf{x}_c^i - \delta\mathbf{x}_c^j\|_2, \tag{6}$$

where i, j are the indices of nearby control points. We also limit the Gaussian scale to prevent the Gaussians being too large:

$$\mathcal{L}_{scale} = \sum_{i=1}^{N} \max(0.01, \mathbf{s}^i). \tag{7}$$

The final loss is

$$\mathcal{L} = \mathcal{L}_1 + 0.1\mathcal{L}_{lpips} + 0.1\mathcal{L}_{ctrl} + \mathcal{L}_{scale}. \tag{8}$$

Testing. Following AnimatableGaussians [28], we use PCA to project novel poses to the space of training poses. Specially, all pose vectors of training poses form a matrix X. we perform PCA on X and then project the novel pose to the linear space before using it as input. This strategy can make the model generalize better on the novel pose by fitting it within the space of the training poses.

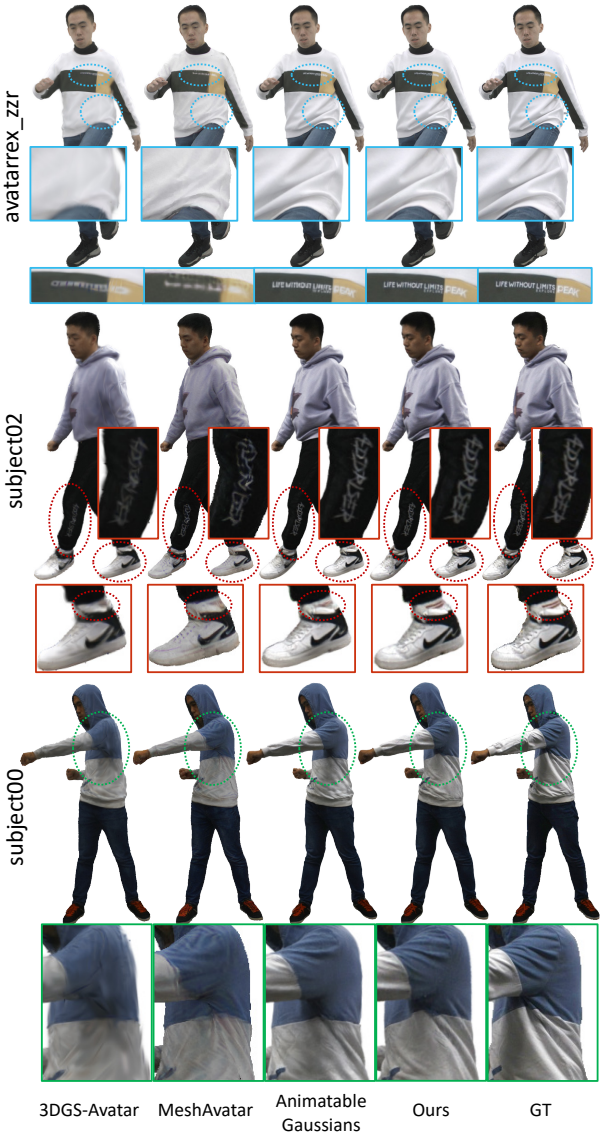

Figure 4. Qualitative comparison with the state-of-the-art methods on training pose reconstruction (top two subjects) and novel pose synthesis (bottom subject).

4. Experiments

We conduct experiments on the following public datasets:

AvatarRex [59]. This dataset contains full-body videos captured at 2K resolution from 16 views. We use 3 sequences and 14 views for our experiments.

THuman4.0 [58]. The dataset contains videos of people with rich wrinkles in their appearance. The videos are captured at 1K resolution. We use 3 sequences and 24 views for our experiments.

ActorsHQ [15]. ActorsHQ is a high-quality dataset. We use 7 sequences and select 39 full-body views for training. We use 4x down-sampled images, with each image approximately 1K resolution.

Each sequence of the above datasets contains 1000 to 2000 frames. We also use the SMPL-X registrations provided by AnimatableGaussians [28] for each sequence.

For quantitative experiments, we use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Learned Perceptual Image Patch Similarity (LPIPS [56]), and Frechet Inception Distance (FID [12]) as metrics. We calculate PSNR, SSIM, and FID on the whole image, and LPIPS within the body bounding box. Training time and rendering speed are all evaluated on a single NVIDIA 3090 card.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
3DGS-Avatar [42]	28.9530	0.9741	0.0464	24.9938
MeshAvatar [4]	29.3154	0.9731	0.0397	19.7409
AnimatableGaussians [28]	31.2992	0.9831	0.0251	11.3421
Ours	**32.7456**	**0.9868**	**0.0226**	**10.1169**

Table 1. Quantitative comparison with the state-of-the-art methods under training poses.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
PoseVocab [27]	26.3784	0.9707	0.0592	49.4541
3DGS-Avatar [42]	27.5524	0.9737	0.0597	31.0979
MeshAvatar [4]	27.4025	0.9717	0.0571	27.4278
AnimatableGaussians [28]	28.1106	0.9741	0.0552	19.2324
Ours	**28.3263**	**0.9747**	**0.0537**	**19.0217**

Table 2. Quantitative comparison with the state-of-the-art methods under novel poses.

Method	Training Time (hours) ↓	FPS ↑
AnimatableGaussians [28]	100	10
DEGAS [43]	55	30
Ours	**17.5**	**166**

Table 3. Performance comparison. All methods are trained for 800K iterations. The training time and rendering fps under novel poses are recorded on a single NVIDIA 3090 card. For DEGAS [43], we directly adopt the data from the original paper.

4.1. Comparison

Quality. We compare our method with 3DGS-Avatar [42], MeshAvatar [4], and AnimatableGaussians [28]. We conduct experiments on sequence avatarrex_zzr from AvatarReX dataset and sequence subject00, subject02 from THuman4.0 dataset. avatarrex_zzr and subject02 are evaluated under training poses, while subject00 is evaluated under novel poses. We present qualitative results in Fig. 4. In the cases of avatarrex_zzr and subject02, 3DGS-Avatar and MeshAvatar struggle to learn fine wrinkles and textures. Although both our method and AnimatableGaussians can capture wrinkles similar to the ground truth, our method captures better details (e.g., the text "LIFE WITHOUT LIMITS" on the chest of avatarrex_zzr and the socks of subject02). In subject00, 3DGS-Avatar and MeshAvatar are not able to render high-quality appearances under novel poses, while our method and AnimatableGaussians can produce details like wrinkles. Therefore, our method surpasses other state-of-the-art methods like 3DGS-Avatar and MeshAvatar, and achieves comparable results to AnimatableGaussians, but with higher-fidelity details. For quantitative results, Tab. 1 presents the metric comparison under training poses and Tab. 2 under novel poses. Our method achieves the best results, demonstrating our method has strong learning ability in modeling human avatars and generalizes well to novel poses.

Speed. We also compare the training and rendering speed with AnimatableGaussians [28] and DEGAS [43], which could render high-quality human avatars as well. Since DEGAS hasn't released its code, we directly use the performance data from its paper. Tab. 3 shows the training time and rendering speed. Our method uses less time to train and significantly outperforms other methods in rendering speed. This is because these methods use a large CNN to predict Gaussian properties, requiring a lot of time to go through the network (i.e., AnimatableGaussians takes 107ms to infer their network once). In contrast, our method takes only about 6ms to render a frame for 200K Gaussians, with 1.5ms for obtaining Gaussian coefficients, 3.3ms for combining the basis and LBS, and 1.1ms for rasterization, making the rendering process highly efficient.

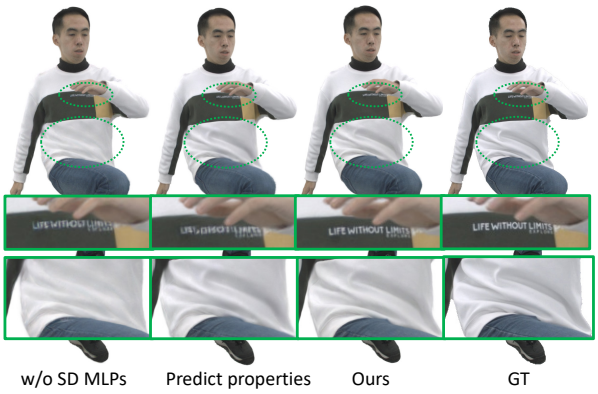

Figure 5. Qualitative comparison of several design choices.

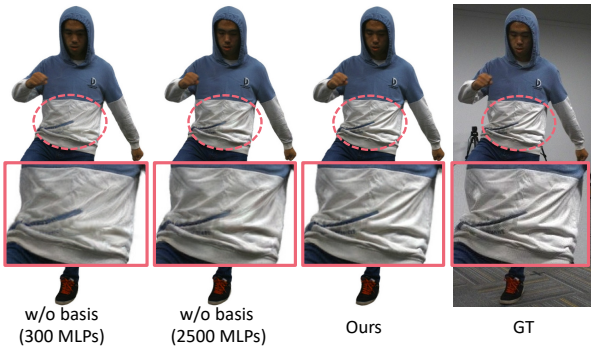

Figure 6. Ablation study on Gaussian offset basis.

4.2. Ablation Study

In this section, we evaluate several of our key designs. These designs impact the results in terms of quality or efficiency.

Spatially Distributed MLPs. Using multiple spatially distributed MLPs enhances the model's learning ability. For comparison, we use a single MLP to output the coefficient to combine the basis. Tab. 4 "w/o SD MLPs" shows that this approach greatly decreases the metrics, proving that a single MLP lacks sufficient learning capacity. Fig. 5 "w/o SD MLPs" also demonstrates that a single MLP is inadequate for capturing appearance details.

We also evaluate how the number of spatially distributed

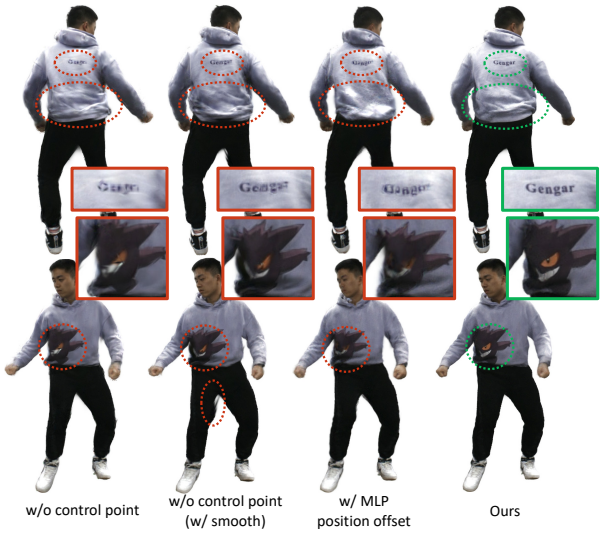

Figure 7. Ablation study on control point.

	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	Training ↓	FPS ↑
w/o SD MLPs	30.1941	0.9791	0.0314	15.4545	15.8 h	182
w/o basis (300 MLPs)	31.8444	0.9832	0.0294	14.4292	**15.7 h**	**205**
w/o basis (2500 MLPs)	32.3912	0.9854	0.0244	11.0721	27.0 h	115
Predict properties	31.9340	0.9834	0.0260	10.9757	24.2 h	51
w/o control point	32.6780	0.9859	0.0241	10.6394	17.8 h	161
w/ MLP position offset	31.8514	0.9816	0.0338	17.8664	18.4 h	148
Ours	**32.7456**	**0.9868**	**0.0226**	**10.1169**	17.5 h	166

Table 4. Ablation study on design choices.

MLP number	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	Training ↓	FPS ↑
50	32.6625	0.9863	0.0237	11.1197	16.6 h	173
300 (Ours)	32.7456	0.9868	0.0226	10.1169	17.5 h	166
800	32.7313	0.9866	0.0224	9.8885	20.8 h	149

Table 5. Quantitative comparison of different number of spatially distributed MLPs.

Basis number	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	Training ↓	FPS ↑
5	32.3523	0.9854	0.0248	11.5850	16.5 h	175
15 (Ours)	32.7456	0.9868	0.0226	10.1169	17.5 h	166
40	32.7154	0.9866	0.0221	10.0522	20.4 h	156
40 (large MLP)	32.7010	0.9869	0.0220	9.9707	24.6 h	140

Table 6. Quantitative comparison of different number of the basis.

MLPs will influence the results. Tab. 5 shows the quality metrics and speed. To balance quality and efficiency, we choose to use $F = 300$ MLPs.

Gaussian Offset Basis. We demonstrate that high-frequency details are difficult to recover without Gaussian offset basis. To prove this, we let the spatially distributed MLPs output Gaussian property offsets, and perform interpolation on the Gaussian property offsets. We experiment this idea using 300 and 2500 MLPs. Fig. 6 "w/o basis (300 MLPs)" and "w/o basis (2500 MLPs)" cannot recover all the fine details. In comparison, our results are almost identical to the ground truth. For quantitative comparison, Tab. 4 "w/o basis (300 MLPs)" and "w/o basis (2500 MLPs)" show a decline in metrics compared with ours. Also, increasing the number of MLPs to 2500 can also reduce rendering speed, as shown in Tab. 4 "w/o basis (2500 MLPs)".

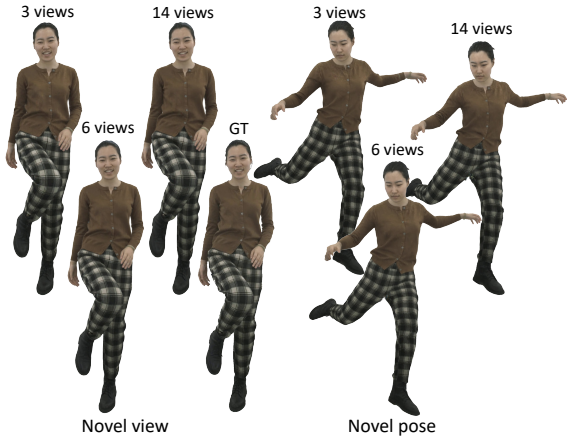

Figure 8. Results of different view number.

We also shows that other possible designs are not as good as our basis combination. For example, following Ye et al. [54], we assign a learnable vector to each Gaussian. Each Gaussian finds its nearest spatially distributed MLP and the MLP takes the learnable vector and pose vector as input and directly predicts Gaussian property offsets. Fig. 5 "Predict properties" shows the results, indicating that this design still fails to capture very high-frequency details (such as the text on the chest). Additionally, because the MLPs take the pose and learnable vector as input, they need to be inferred many times, thus the training time increases and rendering speed is significantly reduced, as shown in Tab. 4 "Predict properties".

We also verify the suitable number of basis. Tab. 6 shows quantitative results for 5, 15, and 40 bases. The results of 15 and 40 bases are comparable with tiny gaps. A reasonable guess is that the linear sub-space of the Gaussian property offsets space can already be well supported with 15 bases. Increasing to 40 bases does not necessarily improve the representation capability. To further validate the guess, we use larger MLPs to learn 40 coefficients for 40 bases, in case the performance is limited by the MLP capacity, but the results are still comparable with those of 15 bases with slight differences (see Tab. 6). Empirically, we choose $B = 15$ as our selection.

Control Point. Without control points, Gaussians can move freely, resulting in suboptimal results when animated under novel pose. Fig. 7 "w/o control point" shows qualitative result. Even with additional local smoothness constraints (Fig. 7 "w/o control point (w/ smooth)"), Gaussians still cannot be well-constrained to the surface, causing details such as text and textures to become corrupted under novel pose. We also demonstrate that utilizing another MLP to predict the position offsets for each control point is still insufficient to render good results (Fig. 7 "w/ MLP position offset"). This is because using an MLP to learn position offsets for all control points in a sequence may still exceed

Figure 9. Our method achieves high-quality human avatar reconstruction and animation under novel poses.

one MLP's learning capacity. Tab. 4 "w/ MLP position offset" also shows decline in metrics.

Sparse View. Our method can be trained under sparse viewpoints. We evaluate it using sequence avatarrex_lbn2 from AvatarRex dataset, and present results in Fig. 8 using 3, 6 and 14 viewpoints. Our method achieves high-fidelity results even with only three viewpoints for training and can still be animated under novel poses.

4.3. More Results

We show more results in Fig. 9. The presented results are animated under novel poses. Our method achieves high-quality human avatar reconstruction and animation. We also provide a viewer that allows users to interactively animate the reconstructed human avatar. Please refer to the supplementary video for more visual results. We note that fps in the viewer is slightly lower due to additional data transmission overhead.

5. Conclusion and Limitation

In this paper, we propose a method capable of modeling high-fidelity human avatars with high-frequency details while the rendering speed is very fast. We use the spatially distributed MLPs to infer the coefficients for the Gaussian offset basis. The smoothly interpolated coefficients combined with freely learned basis can produce distinctly different Gaussian property offsets, allowing the ability to learn high-frequency details. We also use control points to constrain the Gaussians to be distributed on a surface layer without moving inside the body. Experiments demonstrate that our method surpasses previous state-of-the-art methods both in reconstruction fidelity and rendering performance. Currently our avatar appearance is conditioned on pose and cannot model other complex cloth dynamics such as long skirt swaying in the wind. Modeling clothes as a separate layer and incorporating simulation could potentially improve the applicability of our model. Reconstructing human avatars with high-fidelity pose-dependent appearances from monocular videos is another direction worth exploring. Our method still relies on multi-view capture, pose estimation, and template mesh extraction, which makes the pipeline quite heavy. In future we plan to use fewer RGBD cameras to reduce the complexity of pipeline, as the difficulties of skeleton estimation and mesh reconstruction can be largely reduced with the depth information.

Acknowledgment

The authors would like to thank the reviewers for their insightful comments. This work is supported by the National Key Research and Development Program of China (No.2022YFF0902302), NSF China (No. 62322209 and No. 62421003), the gift from Adobe Research, the XPLORER PRIZE, and the 100 Talents Program of Zhejiang University. The source code is available at https://gapszju.github.io/mmlphuman.

References

[1] Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabián 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. 2

[2] Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 2300–2308, 2015. 2

[3] Yue Chen, Xuan Wang, Xingyu Chen, Qi Zhang, Xiaoyu Li, Yu Guo, Jue Wang, and Fei Wang. Uv volumes for real-time rendering of editable free-view human performance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16621–16631, 2023. 2

[4] Yushuo Chen, Zerong Zheng, Zhe Li, Chao Xu, and Yebin Liu. Meshavatar: Learning high-quality triangular human avatars from multi-view videos. *arXiv preprint arXiv:2407.08414*, 2024. 3, 6

[5] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 2

[6] Xiang Deng, Zerong Zheng, Yuxiang Zhang, Jingxiang Sun, Chao Xu, Xiaodong Yang, Lizhen Wang, and Yebin Liu. Ram-avatar: Real-time photo-realistic avatar from monocular videos with full-body control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1996–2007, 2024. 2

[7] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics (TOG)*, 41(6):1–12, 2022. 3

[8] Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Learning neural volumetric representations of dynamic humans in minutes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8770, 2023. 2

[9] Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12858–12868, 2023. 2

[10] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2):1–17, 2019. 2

[11] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *ACM Transactions on Graphics (ToG)*, 40(4):1–16, 2021. 2

[12] 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

[13] Hezhen Hu, Zhiwen Fan, Tianhao Wu, Yihan Xi, Seoyoung Lee, Georgios Pavlakos, and Zhangyang Wang. Expressive gaussian human avatars from monocular rgb video. *arXiv preprint arXiv:2407.03204*, 2024. 3

[14] Shoukang Hu, Tao Hu, and Ziwei Liu. Gauhuman: Articulated gaussian splatting from monocular human videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20418–20431, 2024. 3

[15] 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. 3, 5

[16] 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. 3

[17] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16922–16932, 2023. 2

[18] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *European Conference on Computer Vision*, pages 402–418. Springer, 2022. 2

[19] Yujiao Jiang, Qingmin Liao, Xiaoyu Li, Li Ma, Qi Zhang, Chaopeng Zhang, Zongqing Lu, and Ying Shan. Uv gaussians: Joint learning of mesh deformation and gaussian textures for human avatar modeling. *arXiv preprint arXiv:2403.11589*, 2024. 3

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2, 5

[21] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 505–515, 2024. 3

[22] Youngjoong Kwon, Lingjie Liu, Henry Fuchs, Marc Habermann, and Christian Theobalt. Deliffas: Deformable light fields for fast avatar synthesis. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[23] Youngjoong Kwon, Baole Fang, Yixing Lu, Haoye Dong, Cheng Zhang, Francisco Vicente Carrasco, Albert Mosella-Montoro, Jianjin Xu, Shingo Takagi, Daeil Kim, et al. Generalizable human gaussians for sparse view synthesis. In *European Conference on Computer Vision*, pages 451–468. Springer, 2025. 3

[24] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19876–19887, 2024. 3

[25] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *European Conference on Computer Vision*, pages 419–436. Springer, 2022. 2

[26] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6):194–1, 2017. 3

[27] Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. Posevocab: Learning joint-structured pose embeddings for human avatar modeling. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 2, 6

[28] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19711–19722, 2024. 1, 2, 3, 4, 5, 6

[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 transactions on graphics (TOG)*, 40(6):1–16, 2021. 2

[30] Yuxiao Liu, Zhe Li, Yebin Liu, and Haoqian Wang. Texvocab: Texture vocabulary-conditioned human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1715–1725, 2024. 2

[31] Shugao Ma, Tomas Simon, Jason Saragih, Dawei Wang, Yuecheng Li, Fernando De La Torre, and Yaser Sheikh. Pixel codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 64–73, 2021. 2

[32] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. 3

[33] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1, 2

[34] Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. Expressive whole-body 3d gaussian avatar. *arXiv preprint arXiv:2407.21686*, 2024. 3

[35] Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 788–798, 2024. 1, 3

[36] 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

[37] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. Ash: Animatable gaussian splats for efficient and photoreal human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1165–1175, 2024. 3

[38] Pramish Paudel, Anubhav Khanal, Ajad Chhatkuli, Danda Pani Paudel, and Jyoti Tandukar. ihuman: Instant animatable digital humans from monocular videos. *arXiv preprint arXiv:2407.11174*, 2024. 3

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

[40] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. 2

[41] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021. 2

[42] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5020–5030, 2024. 1, 3, 6

[43] Zhijing Shao, Duotun Wang, Qing-Yao Tian, Yao-Dong Yang, Hengyu Meng, Zeyu Cai, Bo Dong, Yu Zhang, Kang Zhang, and Zeyu Wang. Degas: Detailed expressions on full-body gaussian avatars. *arXiv preprint arXiv:2408.10588*, 2024. 3, 6

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

[45] David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. Haha: Highly articulated gaussian human avatars with textured mesh prior. *arXiv preprint arXiv:2404.01053*, 2024. 3

[46] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics*, 18(4):643–650, 2012. 2

[47] Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. Styleavatar: Real-time photo-realistic portrait avatar from a single

video. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2

[48] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *European conference on computer vision*, pages 1–19. Springer, 2022. 2

[49] Jing Wen, Xiaoming Zhao, Zhongzheng Ren, Alexander G Schwing, and Shenlong Wang. Gomavatar: Efficient animatable human modeling from monocular video using gaussians-on-mesh. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2059–2069, 2024. 3

[50] 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*, pages 16210–16220, 2022. 2

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

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

[53] Donglai Xiang, Fabian Prada, Zhe Cao, Kaiwen Guo, Chenglei Wu, Jessica Hodgins, and Timur Bagautdinov. Drivable avatar clothing: Faithful full-body telepresence with dynamic clothing driven by sparse rgb-d input. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 2

[54] Keyang Ye, Tianjia Shao, and Kun Zhou. Animatable 3d gaussians for high-fidelity synthesis of human motions. *arXiv preprint arXiv:2311.13404*, 2023. 1, 3, 7

[55] Zhengming Yu, Wei Cheng, Xian Liu, Wayne Wu, and Kwan-Yee Lin. Monohuman: Animatable human neural field from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16943–16953, 2023. 2

[56] 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. 5

[57] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19680–19690, 2024. 3

[58] 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*, pages 15893–15903, 2022. 2, 3, 5

[59] 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): 1–19, 2023. 2, 3, 5