

High-Fidelity Mobile Avatars with Pruned Local Blendshapes

Youyi Zhan^{1,2†} He Wang^{2,3} Tianjia Shao^{1,4*} Kun Zhou^{1,4}

¹State Key Lab of CAD&CG, Zhejiang University ²University College London

³AI Centre, University College London

⁴Hangzhou Research Institute of Holographic and AI Technology



Figure 1. Our method creates high-fidelity human avatars from multi-view video. The avatar can run on multiple platforms, including desktop PCs and mobile devices with ultra-fast rendering speed (1683 FPS on an RTX 4090 GPU and 120 FPS on mobile devices).

Abstract

We propose a method to reconstruct high-fidelity human avatars from multi-view video that can run on mobile devices. Many works can model high-quality Gaussian-based full-body avatars from multi-view video. However, these methods require heavy computation to obtain pose-dependent appearance, making deployment on mobile devices very difficult. Recent methods distill from pretrained models and model pose-dependent nonlinear Gaussian attributes by linearly combining global pose features with blendshapes. Although they can run on mobile devices, they suffer some loss of detail. We observe that nearby Gaussians are often highly correlated within a local region of the body, and can be linearly modeled with less error. Therefore, we use local linear blendshapes in small body parts to capture global nonlinear changes of Gaussian attributes. To further reduce computation and model size, we propose to remove blendshapes for Gaussians whose attributes change little, yielding a minimal blendshape representation. Our method is an end-to-end training method without a pretrained model. To make it run on multiple devices, we implement our method using WebGPU. Experi-

ments show that our method can render high-quality human avatars with better details, and can reach 120 FPS at 2K resolution on mobile devices. Our project page is at <https://gapszju.github.io/webavatar/>.

1. Introduction

The need for high-fidelity 3D human avatar is universal in computer graphics, initially on powerful hardware such as desktop computers and rapidly moving onto portable devices including augmented/virtual reality, mobile devices. Recently, 3D Gaussian Splatting (3DGS [28]) appears as a new tool for fast construction of realistic human avatars from real images/videos, but it has been mainly used in desktop computers where the computational resources, *e.g.* memory, data bus bandwidth, are abundant, and have not so far achieved similar successes on lightweight hardware platforms such as mobile devices.

Recent research has just started to address this limitation [6, 20, 62], mainly by reducing the needed memory and computation. Inspired by SMPL [44] using blendshapes for different poses, TaoAvatar [6] and SqueezeMe [20] uses 3DGS and model the Gaussian attributes as a linear combination of blendshapes and pose features. The main idea behind this is to construct a small basis to represent the distribution of the Gaussian attributes. However, as we show

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

†Work done as a visiting student at University College London

later, their specific strategies are still lacking in terms of training and inference speed and requirements of computation. TaoAvatar and SqueezeMe require pretraining a high-quality large model and then distilling it to a smaller model. Training the original model is slow, as shown by AnimatableGS [39], and distillation further increases the complexity of the method. Also, both methods use a global pose feature for the whole body to linearly combine all the blendshapes. However, given the nonlinearity of the Gaussian attributes with respect to the pose feature, their linear blending can lead to large errors when representing the Gaussian attributes. Finally, blendshapes take significant storage, memory and computation, especially on bandwidth-limited mobile devices. Although SqueezeMe [20] attempts to reduce the computation by sharing the same corrective among close-by Gaussians, it causes severe loss of details.

To make 3DGS fully applicable in human avatars on mobile devices, we propose a new method which does not require pre-training, is not demanding in computational resources, and provides good quality. Our key insight is to exploit the locality of Gaussians. The insight stems from the covariance structure of Gaussian attributes. The attributes of nearby Gaussians are often highly correlated (*e.g.* when an area is occluded, a group of Gaussians may all darken together to simulate shadows). This ‘clustering’ effect is especially pronounced on human body surface, where different parts (*e.g.* arms, legs) correspond to different groups of Gaussians which share strong correlations within the group but not necessarily between groups. From the perspective of Principal Component Analysis (PCA) which treats each Gaussian as one sample (*i.e.* a vector of its attributes), if it is done on all the Gaussian attributes together, it is unlikely for PCA to be able to capture the fine-grained covariance structure as data samples are clustered. Instead, if local PCA can be done, then not only can fine-grained local covariance structures be captured, it also enables us to use a few eigenvectors to capture the total variances. Therefore, we partition the body into several parts and the Gaussian attributes in each part are represented as a combination of local pose features and local blendshapes. Unlike SqueezeMe [20], we do not obtain the blendshapes by distilling a pre-trained model. We allow our model to learn the blendshapes directly during optimization to avoid the pretrain-distillation process.

Very recently, mmlphuman [82] also follows a similar philosophy and explore to use basis functions to represent Gaussian attributes. Despite the similarity, their method is not suitable for mobile devices as it requires large storage and heavy computation. The key reason is their basis functions contains redundant information. Comparatively, despite also using blendshapes as basis functions, we propose a new pruning technique to learn the minimal set of blendshapes. However, it is non-trivial to do so as the ba-

sis is learned, not known a priori. Based on the observation that the dynamic appearance is mainly influenced by a small number of Gaussians, whereas most Gaussians can remain almost unchanged (*e.g.* the appearance of shoes or head stays nearly constant regardless of poses), we can prune the blendshapes for most of the Gaussians, turning them into constant Gaussians. Since we do not know which blendshape should be pruned before training, we train an overparameterized blendshape model, then prune the blendshapes based on the variances of the Gaussian attributes. After further finetuning, this strategy can prune 90% of the blendshape parameters, significantly reducing memory usage and computational cost.

Through exhaustive evaluation, we demonstrate that our method incurs small computational costs, achieving high-quality rendering, and outperform the existing methods. We implement our method using WebGPU, allowing it to run directly in a web browser as well as on mobile devices. Experiments show that our method is able to model challenging poses and complex appearances, and can achieve 2K resolution at 120 FPS on mobile devices.

Our contributions can be summarized as follows:

- We propose a new 3DGS method in human avatars which achieves 2K resolution at 120 FPS on mobile devices.
- We propose a new pruning strategy to minimize the size of the model while learning leading to a massive speed-up in inference.
- We analyze the local covariance structure of 3DGS for human avatars, highlighting and provide insights in how to utilize the locality for reducing model size while retaining good quality.
- We open-source our implementation. To the best of our knowledge, this is the first open-source full-body human 3DGS method which is capable of real-time and high-quality, pose-dependent appearance rendering on mobile hardware.

2. Related Work

2.1. High-fidelity Avatar Rendering

High-quality human avatars have been explored for a long time. Earlier methods typically rely on mesh with texture mapping [4, 9, 15, 65]. However, mesh-based representation usually struggles to model complex appearance and geometry. Differentiable volumetric rendering techniques, such as NeRF [48, 50], enable researchers to model high-fidelity avatar appearance directly from video. Previous works like Nerfies [54], HyperNeRF [55], and Li et al. [37] model avatar deformations using MLP-based deformation fields. Subsequent methods use tri-plane [73, 76] or voxel [40, 66] or transformer [63] to capture high-frequency space-varying information and render high-fidelity human appearance. 3D Gaussian Splatting (3DGS [28, 74]) pro-

vides a new representation for avatar modeling. Several methods [31, 41, 83] utilize Gaussian rendering to produce high-quality appearance. [10, 24, 25, 69, 77] can render dynamic scene and represent the avatars in volumetric video format. Although these methods can reconstruct high-quality avatars, they primarily focus on reconstruction and the avatars are not animatable.

2.2. Animatable Avatar Rendering

More works focus on reconstructing avatars that can be controlled and animated with given poses. To get realistic rendering, it is essential to model pose-dependent appearance for avatars. Some works [2, 16] attempt to model animatable avatars by attaching predicted textures to a deformable mesh to model pose-dependent appearance. NeRF [48] is also popular to model animatable avatars in recent years [12, 13, 22, 23, 30, 35, 42, 43, 45, 47, 57, 58, 67, 71, 79, 80, 84]. These methods typically map points from the observation space to the canonical space and predict color and opacity with input pose as the condition. [21, 38, 85] further propose pose encoding methods, enabling high-fidelity and dynamic human appearance. Although high-quality, volumetric representation is slow to render. Even some methods [7, 11] use neural textures to achieve real-time rendering, they still require substantial compute resources. 3D Gaussian Splatting (3DGS [28]) achieves higher-quality and faster rendering by rasterizing Gaussians in space. Many methods achieve realistic and real-time results using 3DGS [14, 17–19, 29, 32, 33, 49, 59, 70, 75, 81]. They typically predict Gaussian attributes with the pose as input to model animatable avatars. Some methods [8, 27, 39, 61, 68] further use convolutional networks to decode Gaussian attributes to model dynamic appearance and therefore achieve higher rendering quality compared to other methods. However, these methods still require heavy computation to decode Gaussian attributes. Even though they can run in real time on a desktop GPU, it is still difficult to transfer their models to mobile devices with limited memory bandwidth and computation capacity.

2.3. Efficient Avatar Rendering

Improving avatar rendering speed is critical for applications, especially on mobile devices. MoRF [3] uses deferred neural rendering to run on mobile at low resolution and low frame rate. SplattingAvatar [60] attaches Gaussians to mesh template and implements the method on mobile devices, which can render at 30 FPS. However, it does not model dynamic appearance, resulting in suboptimal results. HRM²Avatar [62] focuses on monocular capture and mobile running, and therefore cannot model rich dynamic appearance either. Some works recognize the trade-off between decoding dynamic Gaussian attributes and computational cost and seek a balance between the two. A common

design is to decode Gaussian attributes using simple blendshape computations. Some recent works [1, 6, 20, 34, 46, 82, 86] follow this idea. In particular, ScaffoldAvatar [1] uses patch-based local expression features for face avatars, but focuses only on the head region. SqueezeMe [20] and TaoAvatar [6] distill smaller models from convolutional network outputs. Their compact model is expressed by linear blendshapes and can run high-quality avatars in real time on VR headsets. However, they rely on pre-trained models and still fail to capture some details because they use global pose features to linearly express nonlinear Gaussian attributes. Mmlphuman [82] predicts features for each Gaussian to model nonlinear changes, but its heavy blendshapes take much storage and limit the rendering speed. The concept of local blendshapes has been explored in prior facial animation and deformation works [5, 26, 51, 52, 64, 72]. We propose a method with local linear features and a compact blendshape representation that can run high-fidelity animatable full-body avatars on mobile devices.

3. Method

Our method reconstructs avatars from multi-view videos, which then can be rendered with high-fidelity appearances under different poses. Similar to previous works [6, 20, 39, 82], we first extract the masks of the body and track the SMPL-X [44] skeleton in each frame. We also employ a canonical template mesh for an avatar. The reconstruction contains two stages. In the first stage (Sec. 3.1), we partition the body into multiple parts, and linearly combine the local pose features and the blendshapes to obtain the Gaussian attributes. In the second stage (Sec. 3.2), we prune the blendshapes to remove as much as redundancy, reducing computational and memory costs. In Sec. 3.3–Sec. 3.4, we provide implementation details for training and for running the model on mobile devices.

3.1. Local Linear Blendshapes

Our avatar is represented as N_g Gaussians uniformly distributed on the template mesh. Each Gaussian includes a set of neutral attributes, including rotation \mathbf{r} , scale \mathbf{s} , color \mathbf{c} , opacity o , and position \mathbf{x} . To get pose-dependent appearance and model non-rigid deformation, we predict Gaussian correctives using local linear blendshapes for a given pose θ_p and expression θ_e . Specifically, the body template is partitioned to N_G parts, and Gaussians are grouped to body parts depending on the positions. We follow the position blendshapes of SMPL [44] and define a local blendshape matrix B^i for the i th body part. B^i has shape $[N_{G^i}, N_B, 10]$, where N_{G^i} is the number of Gaussians belonging to the i th part, N_B is the dimension of the blendshape, and 10 represents parameters including rotation in quaternion form, scale and RGB color. Given a local pose feature $\mathbf{e}^i \in \mathbb{R}^{N_B}$, the correctives of rotation, scale and

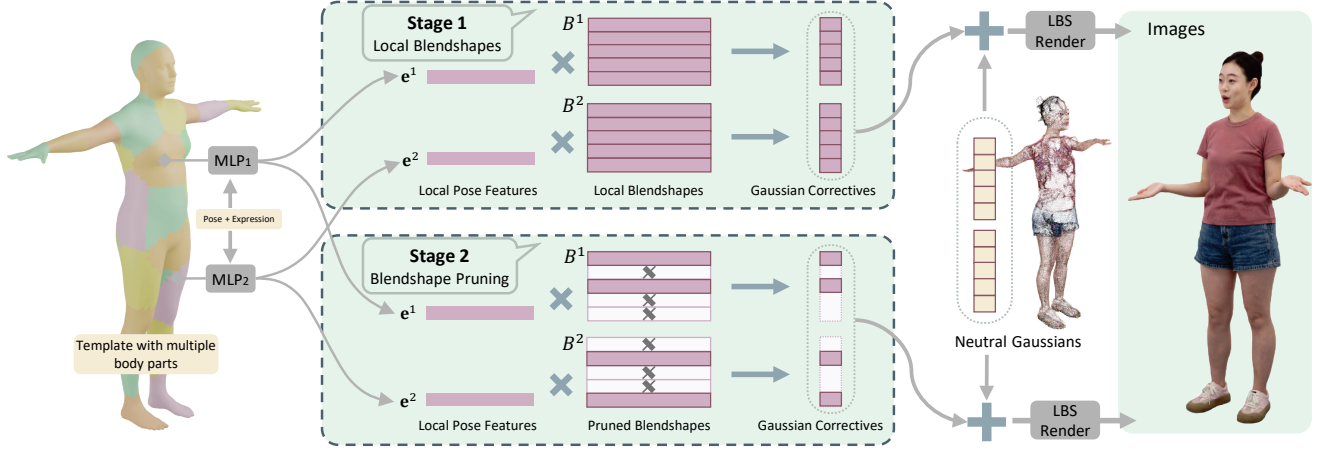


Figure 2. Pipeline overview. We first partition the template body to multiple parts, and predict a local feature for each part. In the first stage, the local feature is combined with local blendshapes to produce the Gaussian correctives, which are then added to neutral Gaussians to model non-rigid appearance changes. In the second stage, most of the blendshapes are pruned, and we apply correctives to only a small number of Gaussians.

color are predicted by summing up the N_B dimension of blendshape B^i ,

$$\{\delta \mathbf{r}^k, \delta \mathbf{s}^k, \delta \mathbf{c}^k | k \in G^i\} = B^i \cdot \mathbf{e}^i, \quad (1)$$

where G^i is the Gaussians in the i th part, and \cdot denotes the weighted summation over the blendshape dimension.

To predict \mathbf{e}^i , we dedicate a small MLP to each body part, which tasks as input the pose and expression, and outputs a local pose feature,

$$\mathbf{e}^i = \text{MLP}^i(\boldsymbol{\theta}_p \oplus \boldsymbol{\theta}_e), \quad (2)$$

where \oplus is the concatenating operation. To reduce the expression influence on the body, we set the pose to zero vector when the body part is not at head.

Inspired by mmlphuman [82], we use control points to predict per Gaussian position correctives. We assume there is a global corrective function over the body and we only need to evaluate this function at a set of discrete nodes. More importantly, the evaluation can be locally computed. Specifically, to determine the nodes, we uniformly sample N_s Gaussians from all the Gaussians and treat them as nodes, so each body part contains some nodes. Next, to enable local evaluation, for the i th body part, we define a local blendshape matrix B_s^i of shape $[N_{G_s^i}, N_B, 3]$ for the nodes belonging to this part, where $N_{G_s^i}$ is the number of nodes in the i th part. As a result, each node’s position corrective $\delta \mathbf{x}_s$ can be obtained by summing up the blendshapes, similar to Eq. (1):

$$\{\delta \mathbf{x}_s^k | k \in G_s^i\} = B_s^i \cdot \mathbf{e}^i. \quad (3)$$

Finally each Gaussian’s position corrective $\delta \mathbf{x}$ is interpolated from the three nearest nodes,

$$\delta \mathbf{x} = \frac{\sum_j \alpha_j \delta \mathbf{x}_s^j}{\sum_j \alpha_j}, \quad (4)$$

where $\alpha_j = 1/\|\mathbf{x} - \mathbf{x}_s^j\|$ is the interpolation coefficient and j is the index of the three nodes nearest to \mathbf{x} .

Similar to TaoAvatar [6], we don’t use high-order SH coefficients to model view-dependent color as the body usually tends to be diffuse. We also do not impose correctives on opacity and let it to be a constant attribute. After applying the correctives, the Gaussians are deformed to posed space and splatted as 2D primitives to render an image with Gaussian rasterizer [28, 78].

3.2. Blendshape Pruning

Although linearly combining the blendshapes to obtain the correctives is effective as it doesn’t involve complicated neural network computation, the model still takes extra storage and memory, because the blendshapes are usually several times as big as the Gaussian attributes. Therefore, we seek a minimal blendshape representation by removing redundant basis functions. We observe that some parts of an avatar change little across different views and poses, such as the shoes. This means the Gaussians at these places do not need pose-dependent correctives and the corresponding blendshapes can be pruned.

However, because we do not know which blendshapes should be kept as we do not know what they are before training, we first train an overparameterized blendshape model and then prune it. Specifically, after the first-stage training, we collect the output correctives of all the training poses and expressions, and compute the variances of rotation, scale and color correctives for each Gaussian,

$$\begin{aligned} \hat{r} &= \text{Var}(\{\delta \mathbf{r}_p\}) \\ \hat{s} &= \text{Var}(\{\delta \mathbf{s}_p\}) \\ \hat{c} &= \text{Var}(\{\delta \mathbf{c}_p\}), \end{aligned} \quad (5)$$

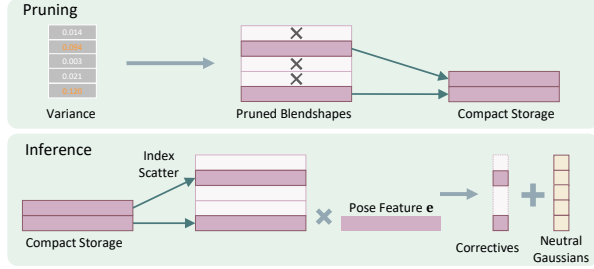


Figure 3. Illustration of blendshape pruning.

where $p \in \Theta$ is the training pose and expression, and Θ contains all the training poses and expressions. $\{\delta\mathbf{r}_p\}, \{\delta\mathbf{s}_p\}, \{\delta\mathbf{c}_p\}$ are the collected attributes of all the poses for a Gaussian. We then keep the blendshape for Gaussians whose attributes exhibit large variance, and prune the blendshape component for Gaussians with small variance. For each attribute type, we independently keep the blendshapes of the top N_P Gaussians with the largest variance and prune the others, where N_P is applied separately to rotation, scale and color, and N_P is a number much smaller than the number of Gaussians N_g . This means the rotation, color, and scale attributes will each retain N_P blendshapes after pruning. This enables us to store the sparse blendshape in a compact form to reduce storage and memory usage. Fig. 3 provides an illustration of the pruning process. We note that the rotation, scale and color blendshapes are pruned separately based on their variances. After this pruning, we further fine-tune the model.

Since the pruning process deletes the correctives and can affect the appearance, we add a constraint on the correctives in the first stage for mitigation. This constraint is to encourage the correctives to be as small as possible, so that the pruning process will not greatly affect the rendering results,

$$\mathcal{L}_{cst} = \lambda_{\delta\mathbf{r}} \|\delta\mathbf{r}\|_1 + \|\delta\mathbf{s}\|_1 + \lambda_{\delta\mathbf{c}} \|\delta\mathbf{c}\|_1 \quad (6)$$

where $\lambda_{\delta\mathbf{r}} = 0.02$, $\lambda_{\delta\mathbf{c}} = 0.002$. During the fine-tuning stage, the loss is not used.

3.3. Implementation Details

At the beginning, we initialize $N_g = 200K$ Gaussians on the template mesh. We use SMPL-X [56] mesh as the template for most of the cases, and construct the template following [39] for loose clothing. To partition the body, we sample $N_G = 256$ points on the template mesh using Poisson-disk sampling, and assign each of the N_g Gaussians to its nearest sampled point. All Gaussians mapped to the same point form one partition. The template mesh is only used for initialization, and once the Gaussians are sampled and grouped to different body parts, we no longer use the mesh. We set $N_G = 256$, $N_B = 16$, $N_s = 10K$ and $N_P = 20K$ in our experiments. The Gaussian position \mathbf{x} is fixed after being sampled on the template mesh, and we

predict $\delta\mathbf{x}$ to compensate the non-rigid displacement under different poses. θ_e consists of the first 10 FLAME [36] expression components and θ_p is 63-dimensional, containing the axis-angle rotations of the main joints.

During training, we employ the same loss function as AnimatableGS [39], which includes L1 loss, LPIPS loss. We also apply a scale loss to prevent the Gaussian from growing too large. The final loss is,

$$\mathcal{L} = \mathcal{L}_1 + \lambda_{l_{lips}} \mathcal{L}_{l_{lips}} + \mathcal{L}_{scale} + \mathcal{L}_{cst}, \quad (7)$$

where $\lambda_{l_{lips}}$ is set to 0.1. The constraint loss \mathcal{L}_{cst} is only applied in the first stage, and will not be used after pruning. We train the first stage for 200K iterations, and second fine-tuning stage for 80K iterations. Training batch size is set to 4. The whole training takes 7.5 hours on an RTX 4090 GPU.

3.4. Running on Mobile Devices

To deploy our model to mobile devices, we quantize all floating-point parameters, including MLP, blendshapes and attributes, to float16, and export some integer parameters to int16. After compression, the model size is just about 19.4 MB and can be easily downloaded on various types of devices. We implement our visualization program using Rust and WebGPU, which can be compiled to a native executable on desktop system (e.g., Windows, macOS and Linux), or a webpage that can be opened on the supported browser. WebGPU is a modern graphics interface that can fully utilize the GPU for rendering and parallel computing. All tests on the mobile devices are conducted by opening the webpage on the browser. We use WebGPU’s compute shaders for MLP inference, combining the blendshapes and animating the avatar with linear blend skinning. Then the Gaussians are rendered using a rasterizer from c3dgs [53]. This rasterizer performs efficient Gaussian sorting and blending using WebGPU. The whole process is fully accelerated by GPU hardware, and can reach 1683 FPS on an RTX 4090 GPU, or 120 FPS on mobile devices with 2K resolution.

4. Experiments

Dataset. We employ four datasets. **AvatarRex** [85]: Each sequence contains videos captured using 16 cameras with 2K resolution. The avatar performs various poses and exhibits rich pose-dependent appearance. **Talking-Body4D** [6]: This dataset contains videos of the avatar standing in front of the cameras and speaking, accompanied by facial expressions and hand gestures. Each sequence contains 60 cameras with 2K resolution. **ActorsHQ** [21]: Each sequence contains about 1,000-2,000 frames and 160 cameras. We use only about 40 cameras. We also follow previous works and use 4x down-sampled images. **DREAMS-Avatar** [61]: It contains videos in



Figure 4. Comparison on AvatarRex dataset. All images are rendered with novel view and novel pose. *The results of SqueezeMe are taken from the appendix of their paper.

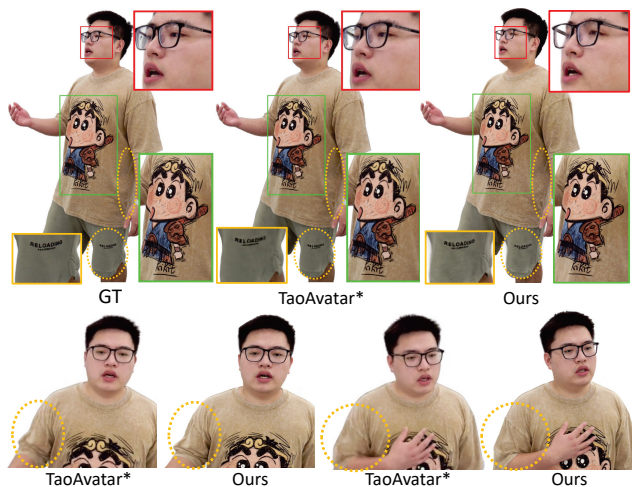


Figure 5. Qualitative comparison with TaoAvatar [6]. *The results of TaoAvatar are taken from their paper and video.

which the avatar performs different poses with rich expressions. Each sequence has 32 viewpoints at 2K resolution and about 1,000–2,000 frames. We used the datasets’ provided SMPL-X registrations for the above datasets. Since ActorsHQ does not contains the skeleton registration, we use the registration provided by AnimatableGS [39].

Platform and Performance Evaluation. We test the performance on three platforms: a laptop with an RTX 3050 GPU, a PC with an RTX 4090 GPU and an Android phone equipped with a Snapdragon 8 Elite chip. Our model can be run using the PyTorch implementation, or the executable

compiled from Rust code. Since our method can run very fast and the python overhead is heavy, to fully utilize platform performance, we use the compiled executable for all performance experiments unless otherwise specified.

4.1. Comparison

We compare with AnimatableGS [39], mmlphuman [82], TaoAvatar [6] and SqueezeMe [20] for high-quality multi-view avatar reconstruction. We emphasize that both TaoAvatar and SqueezeMe do not release the code or model for training or evaluation. Therefore, we follow their setting and take quantitative results and images from their papers for comparison. All results taken from their paper are marked with an asterisk (*).

Since SqueezeMe reports some results on the AvatarRex dataset, we follow SqueezeMe’s experimental setup as closely as possible. Specifically, we test on lbn2 and zrz identities, training with 14 cameras and reserving two cameras for novel-view evaluation. The final 500 frames of each sequence are held out for novel pose evaluation test data. We overlay the rendered avatar on the groundtruth image, and compute L1, PSNR, SSIM, and LPIPS as evaluation metrics. The metrics are only calculated inside the bounding box of the groundtruth masks. We evaluate using novel view and novel poses. We compare with AnimatableGS, mmlphuman and SqueezeMe using this setting, and the results are presented in Tab. 1. AnimatableGS achieves the best novel-pose animation quality but is computationally heavy and slow. Mmlphuman produces results comparable to ours, but its large model size makes mobile deployment

	L1↓	PSNR↑	SSIM↑	LPIPS↓	FPS ₄₀₉₀ ↑	On Mobile
AnimatableGS [39]	0.02270	25.508	0.8655	0.1550	16	✗
mmlphuman [82]	0.02371	25.276	0.8617	0.1573	315	✗
Ours	0.02343	25.346	0.8606	0.1576	1683	✓
SqueezeMe* [20]	0.059	20.051	0.849	0.158	–	✓
TaoAvatar* [6]	–	–	–	–	156	✓

Table 1. Quantitative comparison with high-quality multi-view methods. We experiment on AvatarRex dataset and follow SqueezeMe’s experiment settings. *The results of SqueezeMe [20] and TaoAvatar [6] are taken from their paper.

	Novel View			Novel Pose and Expression		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TaoAvatar* [6]	33.81	0.9689	0.06437	28.38	0.9389	0.08874
Ours	34.44	0.9771	0.03642	27.90	0.9395	0.05582

Table 2. Quantitative comparison with TaoAvatar on Talking-Body4D dataset.

	L1↓	PSNR↑	SSIM↑	LPIPS↓
Ours	0.01488	29.405	0.9207	0.1074
Global feature 16	0.01777	27.645	0.8978	0.1276
Global feature 64	0.01744	27.913	0.8988	0.1270
No pruning	0.01471	29.342	0.9188	0.1076

Table 3. Ablation study on several designs. This table is evaluated on the training pose and novel view on AvatarRex dataset.

challenging. Under the same experimental settings, our results outperform SqueezeMe. We also provide qualitative results in Fig. 4. Our results are comparable to SqueezeMe, but with more details in some cases (red circles), and are also comparable to mmlphuman and AnimatableGS.

TaoAvatar [6] evaluates on TalkingBody4D. Therefore, we also run experiments on the same dataset. We use four sequences. For each sequence we hold out one camera for testing and train with the remaining cameras. The selected sequence and test viewpoint match those in the figure 3 in TaoAvatar paper. In each sequence we also reserve last 200 frames for novel pose testing. Metrics are computed over the entire image. Tab. 2 shows the quantitative results. Our method achieves better results for most of the metrics. Fig. 5 also shows novel-view comparisons with TaoAvatar. The images are picked from its paper and supplementary video. Our method is able to captures more details, such as small text on the pants and wrinkles on the clothing. In summary, among the methods that can run on mobile devices (TaoAvatar and SqueezeMe), our method can produce better results on the same data case.

We also present the performance of the methods in Tab. 1. The rendering resolution is 2K. Our design can decode Gaussian attributes under different poses in very little time and can run extremely fast on a 4090 GPU, outperforming other methods in efficiency.

4.2. Ablation Study

Local Linear Blendshape. Both SqueezeMe [20] and TaoAvatar [6] use a global pose feature to combine blendshapes to obtain Gaussian attributes. We show that mod-

	FPS ₃₀₅₀	GPU Mem(MB)	Model Size(MB)
Ours	312	105	19.5
No Pruning	191	155	72.4

Table 4. Comparison with no pruning design. All render resolution is 2K.

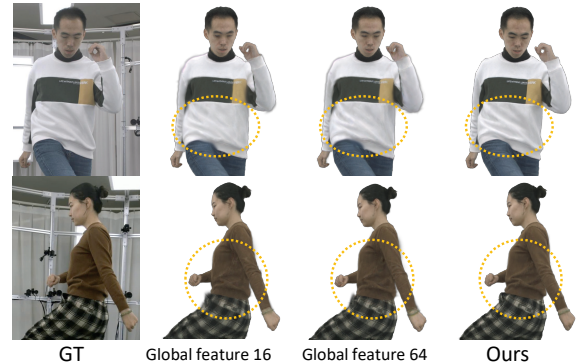


Figure 6. Ablation study on local and global features.



Figure 7. Qualitative comparison with no pruning design. The results of no pruning are almost identical to ours.

eling Gaussian pose-dependent attributes with local linear blendshapes performs better. We use an MLP that takes the pose and expression as input and output a pose feature vector of length 16, which is used to combine all Gaussian blendshapes. We evaluate the design on novel views of AvatarRex dataset. Tab. 3 shows the results. The global pose feature performs far worse. We also test using a pose feature of length 64, which is the same setting used by SqueezeMe, and the results won’t get better. Fig. 6 also shows that using a global pose feature loses more details. Although SqueezeMe mentions that training the global linear model from scratch performs worse than distilling from a trained model, our method is end-to-end and does not require training a heavy convolutional network like TaoAvatar and SqueezeMe.

Blendshape Pruning. Pruning blendshapes may reduce the model’s learning capacity, since most of the Gaussians’ attribute will be constant and won’t change along with the pose. But we find its impact on rendering results is limited. We compare with a model without the pruning and fine-tuning stage. We note that the first stage model is converged. The quantitative results are given in Tab. 3.



Figure 8. Our method can reconstruct high-fidelity human avatars with various appearance and poses. The avatars can be animated under novel poses.

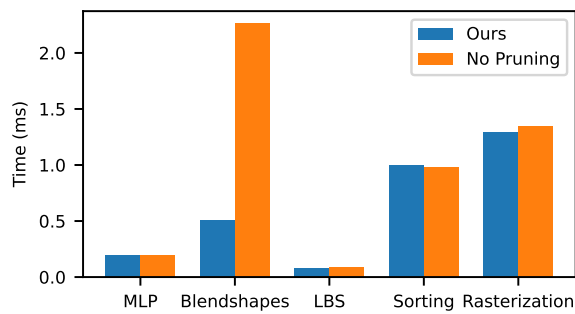


Figure 9. Runtime of each part per frame. The data is collected on a 3050 GPU with 2K rendering resolution.

We show that after pruning and fine-tuning our design can achieve comparable results. This indicates that pruning blendshapes does not severely impair the model’s ability to learn appearance. Fig. 7 also presents the differences between the two designs. Our pruning design produces results similar to the ones without pruning.

Besides delivering comparable quality, our design further reduces model size and improves speed. We report storage and runtime metrics in Tab. 4. Because we prune 90% of the blendshapes, the overall model size and memory footprint are substantially reduced. Although the FPS improvement is not as large as the reduction in model size, we attribute this to a rendering bottleneck. Fig. 9 shows the average time per frame for the two designs. The sorting and rasterization take most of the time. Compared with no pruning, our design still greatly reduces the time of blendshape combination (0.52 ms vs. 2.26 ms).

4.3. Runtime Performance and Application

Fig. 9 shows that compared to the overall time for one frame, the time of decoding Gaussian attributes (MLP + Blendshapes + LBS) is very short. The bottleneck lies in sorting and rendering the Gaussians. Our method can achieve 1683 FPS on 4090 GPU and 120 FPS on mobile phone. Due to browser limitation, the FPS can only reach to the maximum refresh rates of the screen on the phone. Our method decodes Gaussian attributes every frame and can still saturate display refresh rates, demonstrating very high efficiency. Please refer to the supplemental video for the interactive demo. Fig. 8 provides the examples of the avatars reconstructed by our method. We also provide the applications in the supplementary material that allow users to run the digital avatar on their own devices.

5. Conclusion

We propose a lightweight avatar rendering method that can model high-quality dynamic appearance. Compared to previous methods capable of running on mobile devices, our method can be trained end-to-end and does not require pre-training a large convolutional network. We model globally nonlinear Gaussian attributes using local linear modeling to capture finer detail. Blendshape pruning further removes redundant model parameters, increasing speed and reducing model size. We implement our method with WebGPU and run it in the browser. With just a click, a lifelike digital avatar can be brought to anyone, enabling immersive, interactive, and personalized experiences that enhance communication between the physical and virtual worlds.

Acknowledgment

We thank the reviewers for their insightful comments and Doğa Yılmaz for the assistance with this paper. This work is supported in part by NSF China under Grant 62322209 and Grant 62421003, and the XPLOER PRIZE.

References

- [1] Shivangi Aneja, Sebastian Weiss, Irene Baeza, Prashanth Chandran, Gaspard Zoss, Matthias Nießner, and Derek Bradley. Scaffoldavatar: High-fidelity gaussian avatars with patch expressions. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025. 3
- [2] 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. 3
- [3] Renat Bashirov, Alexey Larionov, Evgeniya Ustinova, Mikhail Sidorenko, David Svitov, Ilya Zakharkin, and Victor Lempitsky. Morf: Mobile realistic fullbody avatars from a monocular video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3545–3555, 2024. 3
- [4] 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
- [5] Alan Brunton, Timo Bolkart, and Stefanie Wuhler. Multilinear wavelets: A statistical shape space for human faces. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 3
- [6] Jianchuan Chen, Jingchuan Hu, Gaige Wang, Zhonghua Jiang, Tiansong Zhou, Zhiwen Chen, and Chengfei Lv. Taoavatar: Real-time lifelike full-body talking avatars for augmented reality via 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10723–10734, 2025. 1, 3, 4, 5, 6, 7
- [7] 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. 3
- [8] Yushuo Chen, Zerong Zheng, Zhe Li, Chao Xu, and Yebin Liu. Meshavatar: Learning high-quality triangular human avatars from multi-view videos. In *European Conference on Computer Vision*, pages 250–269. Springer, 2024. 3
- [9] 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
- [10] Pinxuan Dai, Peiquan Zhang, Zheng Dong, Ke Xu, Yifan Peng, Dandan Ding, Yujun Shen, Yin Yang, Xinguo Liu, Rynson WH Lau, et al. 4d gaussian videos with motion layering. *ACM Transactions on Graphics (TOG)*, 44(4):1–14, 2025. 3
- [11] 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. 3
- [12] 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. 3
- [13] 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. 3
- [14] Chen Guo, Junxuan Li, Yash Kant, Yaser Sheikh, Shunsuke Saito, and Chen Cao. Vid2avatar-pro: Authentic avatar from videos in the wild via universal prior. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5559–5570, 2025. 3
- [15] 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
- [16] 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. 3
- [17] Hezhen Hu, Zhiwen Fan, Tianhao Wu, Yihan Xi, Seoyoung Lee, Georgios Pavlakos, Zhangyang Wang, et al. Expressive gaussian human avatars from monocular rgb video. *Advances in Neural Information Processing Systems*, 37:5646–5660, 2024. 3
- [18] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 634–644, 2024.
- [19] 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
- [20] Forrest Iandola, Stanislav Pidrskyi, Igor Santesteban, Divam Gupta, Anuj Pahuja, Nemanja Bartolovic, Frank Yu, Emanuel Garbin, Tomas Simon, and Shunsuke Saito. Squeezeme: Mobile-ready distillation of gaussian full-body avatars. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025. 1, 2, 3, 6, 7
- [21] 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
- [22] 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. 3
- [23] 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. 3
- [24] Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. Hifi4g: High-fidelity human performance rendering via compact gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19734–19745, 2024. 3
- [25] Yuheng Jiang, Chengcheng Guo, Yize Wu, Yu Hong, Shengkun Zhu, Zhehao Shen, Yingliang Zhang, Shaohui Jiao, Zhuo Su, Lan Xu, et al. Topology-aware optimization of gaussian primitives for human-centric volumetric videos. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025. 3
- [26] Pushkar Joshi, Wen C Tien, Mathieu Desbrun, and Frédéric Pighin. Learning controls for blend shape based realistic facial animation. In *ACM Siggraph 2006 Courses*, pages 17–es. 2006. 3
- [27] Hendrik Junkawitsch, Guoxing Sun, Heming Zhu, Christian Theobalt, and Marc Habermann. Eva: Expressive virtual avatars from multi-view videos. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025. 3
- [28] 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, 3, 4
- [29] 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
- [30] 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. 3
- [31] 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
- [32] Changmin Lee, Jihyun Lee, and Tae-Kyun Kim. Mpmavatar: Learning 3d gaussian avatars with accurate and robust physics-based dynamics. *arXiv preprint arXiv:2510.01619*, 2025. 3
- [33] 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
- [34] Linzhou Li, Yumeng Li, Yanlin Weng, Youyi Zheng, and Kun Zhou. Rgbavatar: Reduced gaussian blendshapes for online modeling of head avatars. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10747–10757, 2025. 3
- [35] 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. 3
- [36] 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. 5
- [37] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5521–5531, 2022. 2
- [38] 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. 3
- [39] 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. 2, 3, 5, 6, 7
- [40] Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. High-fidelity and real-time novel view synthesis for dynamic scenes. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–9, 2023. 2
- [41] Hanxi Liu, Yifang Men, and Zhouhui Lian. Creating your editable 3d photorealistic avatar with tetrahedron-constrained gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15976–15986, 2025. 3
- [42] 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. 3
- [43] 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. 3
- [44] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6), 2015. 1, 3
- [45] 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. 3

- [46] 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
- [47] Shaoxu Meng, Tong Wu, Fang-Lue Zhang, Shu-Yu Chen, Yuewen Ma, Wenbo Hu, and Lin Gao. Avatarwild: Fully controllable head avatars in the wild. *Visual Informatics*, 8(3):96–106, 2024. 3
- [48] 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. 2, 3
- [49] Gyeongseok Moon, Takaaki Shiratori, and Shunsuke Saito. Expressive whole-body 3d gaussian avatar. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 3
- [50] 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
- [51] Kyung-Gun Na and Moon-Ryul Jung. Local shape blending using coherent weighted regions. *The Visual Computer*, 27(6):575–584, 2011. 3
- [52] Thomas Neumann, Kiran Varanasi, Stephan Wenger, Markus Wacker, Marcus Magnor, and Christian Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013. 3
- [53] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10349–10358, 2024. 5
- [54] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5865–5874, 2021. 2
- [55] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 2
- [56] 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. 5
- [57] 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. 3
- [58] 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. 3
- [59] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2024. 3
- [60] 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
- [61] 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. In *2025 International Conference on 3D Vision (3DV)*, pages 1529–1540. IEEE, 2025. 3, 5
- [62] Chao Shi, Shenghao Jia, Jinhui Liu, Yong Zhang, Liangchao Zhu, Zhonglei Yang, Jinze Ma, Chaoyue Niu, and Chengfei Lv. HRM²Avatar: High-fidelity real-time mobile avatars from monocular phone scans. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*, Hong Kong, Hong Kong, 2025. Association for Computing Machinery. 1, 3
- [63] Min Shi, Wenke Feng, Lin Gao, and Dengming Zhu. Generating diverse clothed 3d human animations via a generative model. *Computational Visual Media*, 10(2):261–277, 2024. 2
- [64] J Rafael Tena, Fernando De la Torre, and Iain A Matthews. Interactive region-based linear 3d face models. *ACM Trans. Graph.*, 30(4):76, 2011. 3
- [65] 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
- [66] Liao Wang, Kaixin Yao, Chengcheng Guo, Zhirui Zhang, Qiang Hu, Jingyi Yu, Lan Xu, and Minye Wu. Videorf: Rendering dynamic radiance fields as 2d feature video streams. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 470–481, 2024. 2
- [67] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdf. In *European conference on computer vision*, pages 1–19. Springer, 2022. 3
- [68] Shaofei Wang, Tomas Simon, Igor Santesteban, Timur Bagautdinov, Junxuan Li, Vasu Agrawal, Fabian Prada, Shou-I Yu, Pace Nalbene, Matt Gramlich, et al. Relightable full-body gaussian codec avatars. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 3
- [69] Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21750–21760, 2025. 3

- [70] 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](#)
- [71] 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. [3](#)
- [72] Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM transactions on graphics (TOG)*, 35(4):1–12, 2016. [3](#)
- [73] Minye Wu, Zehao Wang, Georgios Kouros, and Tinne Tuytelaars. Tetrirf: Temporal tri-plane radiance fields for efficient free-viewpoint video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6487–6496, 2024. [2](#)
- [74] Tong Wu, Yu-Jie Yuan, Ling-Xiao Zhang, Jie Yang, Yan-Pei Cao, Ling-Qi Yan, and Lin Gao. Recent advances in 3d gaussian splatting. *Computational Visual Media*, 10(4):613–642, 2024. [2](#)
- [75] Wangze Xu, Yifan Zhan, Zhihang Zhong, and Xiao Sun. Sequential gaussian avatars with hierarchical motion context. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13592–13603, 2025. [3](#)
- [76] Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 4k4d: Real-time 4d view synthesis at 4k resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20029–20040, 2024. [2](#)
- [77] Zhen Xu, Yinghao Xu, Zhiyuan Yu, Sida Peng, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Representing long volumetric video with temporal gaussian hierarchy. *ACM Transactions on Graphics (TOG)*, 43(6):1–18, 2024. [3](#)
- [78] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. [4](#)
- [79] Bowei Yin, Junke Zhu, and Zhangjin Huang. Cyclegaussianavatar: Encoding of facial details with the cycle consistency framework. *Visual Informatics*, page 100264, 2025. [3](#)
- [80] 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. [3](#)
- [81] Youyi Zhan, Tianjia Shao, He Wang, Yin Yang, and Kun Zhou. Interactive rendering of relightable and animatable gaussian avatars. *IEEE Transactions on Visualization and Computer Graphics*, 2025. [3](#)
- [82] Youyi Zhan, Tianjia Shao, Yin Yang, and Kun Zhou. Real-time high-fidelity gaussian human avatars with position-based interpolation of spatially distributed mlps. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26297–26307, 2025. [2](#), [3](#), [4](#), [6](#), [7](#)
- [83] 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](#)
- [84] 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. [3](#)
- [85] 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. [3](#), [5](#)
- [86] Wojciech Zielonka, Timo Bolkart, Thabo Beeler, and Justus Thies. Gaussian eigen models for human heads. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15930–15940, 2025. [3](#)