GaussianAvatars: Photorealistic Head Avatars with Rigged 3D Gaussians

Shenhan Qian[1]   Tobias Kirschstein[1]   Liam Schoneveld[2]   Davide Davoli[3]
Simon Giebenhain[1]   Matthias Nießner[1]

[1]Technical University of Munich   [2]Woven by Toyota   [3]Toyota Motor Europe NV/SA
associated partner by contracted service

Figure 1. We introduce GaussianAvatars, a new method to create photorealistic head avatars from multi-view videos. Our avatar is represented by 3D Gaussian splats rigged to a parametric face model. We can fully control and animate our avatars in terms of pose, expression, and view point as shown in the example renderings above.

Abstract

We introduce GaussianAvatars[1], a new method to create photorealistic head avatars that are fully controllable in terms of expression, pose, and viewpoint. The core idea is a dynamic 3D representation based on 3D Gaussian splats that are rigged to a parametric morphable face model. This combination facilitates photorealistic rendering while allowing for precise animation control via the underlying parametric model, e.g., through expression transfer from a driving sequence or by manually changing the morphable model parameters. We parameterize each splat by a local coordinate frame of a triangle and optimize for explicit displacement offset to obtain a more accurate geometric representation. During avatar reconstruction, we jointly optimize for the morphable model parameters and Gaussian splat parameters in an end-to-end fashion. We demonstrate the animation capabilities of our photorealistic avatar in several challenging scenarios. For instance, we show reenactments from a driving video, where our method outperforms existing works by a significant margin.

1. Introduction

Creating animatable avatars of human heads has been a longstanding problem in computer vision and graphics.

In particular, the ability to render photorealistic dynamic avatars from arbitrary viewpoints enables numerous applications in gaming, movie production, immersive telepresence, and augmented or virtual reality. For such applications, it is also crucial to be able to control the avatar, and for it to generalize well to novel poses and expressions.

Reconstructing a 3D representation able to jointly capture the appearance, geometry and dynamics of human heads represents a major challenge for high-fidelity avatar generation. The under-constrained nature of this reconstruction problem significantly complicates the task of achieving a representation that combines novel-view rendering photorealism with expression controllability. Moreover, extreme expressions and facial details, like wrinkles, the mouth interior, and hair, are difficult to capture, and can produce visual artifacts easily noticed by humans.

Neural Radiance Fields (NeRF [28]) and its variants [2, 6, 29] have shown impressive results in reconstructing static scenes from multi-view observations. Follow-up works have extended NeRF to model dynamic scenes for both arbitrary [22, 30, 31] and human-tailored scenarios [13, 18]. These works achieve impressive results for novel view rendering; however, they lack controllability and as such do not generalize well to novel poses and expressions.

The recent 3D Gaussian Splatting method [15] achieves even higher rendering quality than NeRF for novel-view synthesis with real-time performance, optimizing for discrete geometric primitives (3D Gaussians) throughout 3D

---

[1]Project page: https://shenhanqian.github.io/gaussian-avatars

space. This method has been extended to capture dynamic scenes by building explicit correspondences across time steps [27]; however, they do not allow for animations of the reconstructed outputs.

To this end, we propose GaussianAvatars, a method for dynamic 3D representation of human head based on 3D Gaussian splats that are rigged to a parametric morphable face model. Given a FLAME [21] mesh, we initialize a 3D Gaussian at the center of each triangle. When the FLAME mesh is animated, each Gaussian then translates, rotates, and scales according to its parent triangle. The 3D Gaussians then form a radiance field on top of the mesh, compensating for regions where the mesh is either not accurately aligned, or is incapable of reproducing certain visual elements. To achieve high fidelity of the reconstructed avatar, we introduce a binding inheritance strategy to support Gaussian splats without losing controllability. We also explore balancing fidelity and robustness to animate the avatars with novel expressions and poses. Our method outperforms existing works by a significant margin on both novel-view rendering and reenactment from a driving video.

Our contributions are as follows:

- We propose GaussianAvatars, a method to create animatable head avatars by rigging 3D Gaussians to parametric mesh models.
- We design a binding inheritance strategy to support adding and removing 3D Gaussians without losing controllability.

2. Related Work

2.1. Radiance Field Reconstruction

NeRF [28] stores the radiance field of a scene in a neural network and provides photorealistic renderings of novel views with volumetric rendering. Later work [37, 47] represents the scene as voxel grids, achieving comparable rendering quality in a shorter time. Efficiency can be further improved by employing compression methods such as voxel hashing [29] or tensor decomposition [6]. PointNeRF [43] uses point clouds as a scene representation, whereas 3D Gaussian Splatting [15] uses anisotropic 3D Gaussians that are rendered by sorting and rasterization, achieving superior visual quality with real-time rendering. Mixture of Volumetric Primitives [25] uses surface-aligned volumes to achieve fast rendering with high visual fidelity. It also applies residual transformations to primitives. Our method follows 3D Gaussian Splatting [15], benefiting from the expressiveness of anisotropic Gaussians.

A common paradigm to model a dynamic scene is directly storing it in an MLP with 4D coordinates as the input [9, 42] or a 4D tensor with the time dimension or spatial dimensions compressed [1, 4, 7, 36]. These methods can faithfully replay a dynamic scene and produce realistic

novel-view rendering but lack an explicit handle to manipulate the content. Another paradigm learns a static canonical space and the maps time steps back to the canonical space via separate MLPs [30, 31, 34]. Instead of using a deformation MLP, a proxy geometry provides more direct controllability. Liu et al. [24] warp points from an observed space to the canonical space based on the movement of the nearest triangle on an SMPL [26] mesh. Peng et al. [33] deform points with the skeleton of SMPL and neural blending weights. Concurrent works create human body avatars with forward deformation [14, 19, 20, 23, 46] and cage-based deformation [51]. Unlike these methods, we directly attach 3D Gaussians to triangles and explicitly move them, obviating the need for a canonical space and enabling effective mesh finetuning.

2.2. Human Head Reconstruction and Animation

Head avatar creation advanced through the uptake of differentiable rendering and scene representations. Thies et al. [39] instigated a shift toward digital avatars with real-time face tracking and authentic face reenactment. Advancements in image synthesis with neural networks [5, 40] boosted the controllability of head avatars from lip-syncing to expression transfer and head motions [16, 38, 45]. Gafni et al. [8] learn a NeRF conditioned on an expression vector from monocular videos. Grassal et al. [11] subdivide and add offsets to FLAME [21] to enhance its geometry and enable a dynamic texture via an expression-dependent texture field. IMavatar [49] learns a 3D morphable head avatar with neural implicit functions, solving for a map from observed to canonical space via iterative root-finding. HeadNeRF [12] learns a NeRF-based parametric head model with 2D neural rendering for efficiency.

INSTA [52] deforms query points to a canonical space by finding the nearest triangle on a FLAME [21] mesh, and combines this with InstantNGP [29] to achieve fast rendering. Like INSTA, we also use triangles to warp the scene, but we build a consistent correspondence between a 3D Gaussian and a triangle instead of querying the nearest one for each timestep. Zheng et al. [50] explore point-based representations with differential point splatting. They define a point set in canonical space and learn a deformation field conditioned on FLAME's expression vectors to animate the head avatar. While the scale of a point has to be manually set for their method, it is an optimizable parameter for 3D Gaussians. NeRFBlendShape [10] models a dynamic scene by blending hash tables with 3DMM parameters [3, 32]. AvatarMAV [44] decouples motion and appearance, blending voxel grids only for the motion field.

3. Method

As shown in Fig. 2, the input to our method is a multi-view video recording of a human head. For each time step, we

FLAME parameters
shape	$\boldsymbol{\beta}$
vertex offset	$\Delta \boldsymbol{v}$
translation	$\boldsymbol{t}$
joint poses	$\boldsymbol{\theta}$
expression	$\boldsymbol{\psi}$

3D Gaussian parameters
global scaling	$\boldsymbol{s}' = k\boldsymbol{s}$
global position	$\boldsymbol{\mu}' = k\boldsymbol{R}\boldsymbol{\mu} + \boldsymbol{T}$
global rotation	$\boldsymbol{r}' = \boldsymbol{R}\boldsymbol{r}$
parent triangle	i
SH coefficient	$\boldsymbol{h}$
opacity	α

rendering equation

$$\boldsymbol{C} = \sum_{i=1} \boldsymbol{c}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j)$$

loss terms

$$\mathcal{L}_{\text{rgb}} = (1 - \lambda)\mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}}$$

$$\mathcal{L}_{\text{scaling}} = \left\| \max(\boldsymbol{s}, \epsilon_{\text{scaling}}) \right\|_2$$
$$\mathcal{L}_{\text{position}} = \left\| \max(\boldsymbol{\mu}, \epsilon_{\text{position}}) \right\|_2$$

FLAME tracking

input video

① setup local coordinate system for each triangle — triangle scaling k, triangle position $\boldsymbol{T}$, triangle rotation $\boldsymbol{R}$

② assign a 3D Gaussian at the center of each triangle — 3D Gaussian parent triangle i, 3D Gaussian SH coefficients $\boldsymbol{h}$, 3D Gaussian opacity α

③ shift and scale 3D Gaussians during optimization — 3D Gaussian local scaling $\boldsymbol{s}$, 3D Gaussian local position $\boldsymbol{\mu}$, 3D Gaussian local rotation $\boldsymbol{r}$

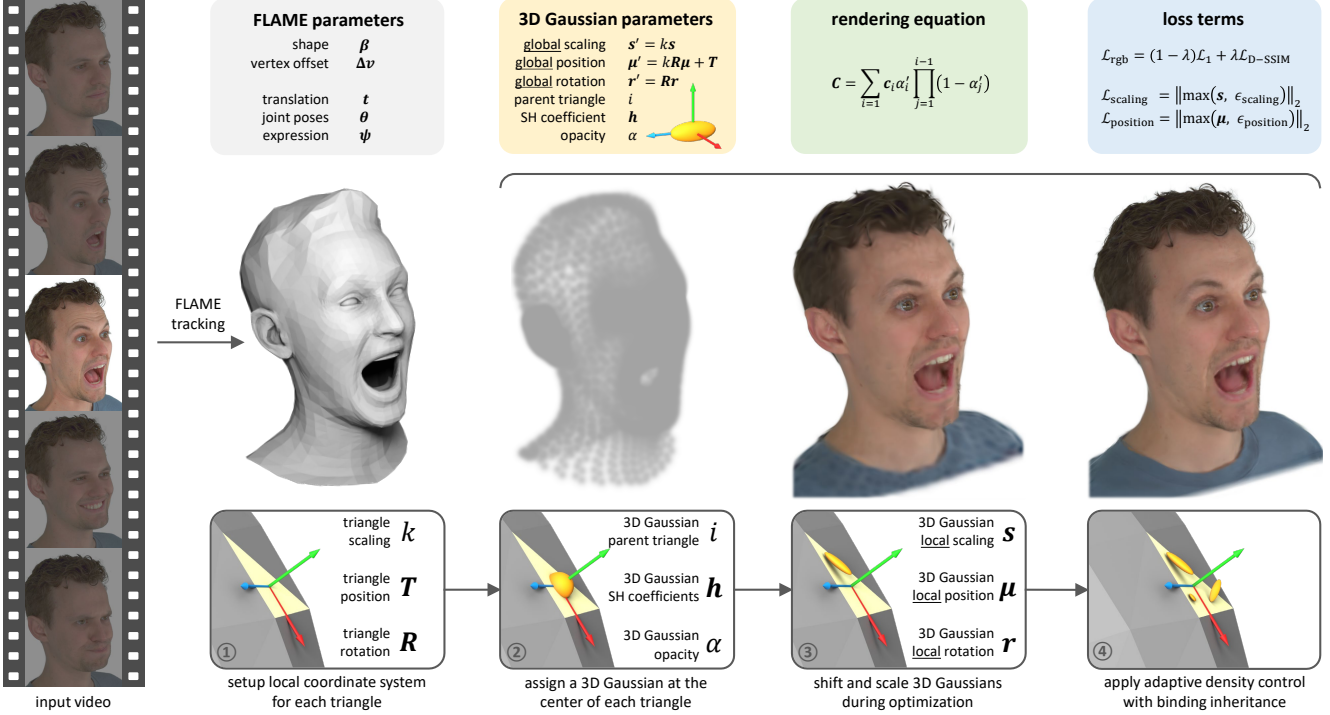④ apply adaptive density control with binding inheritance

Figure 2. **Overview.** Our method binds 3D Gaussian splats to a FLAME [21] mesh locally. We take the tracked mesh for each frame and transform the splats from local to global space before rendering them with 3D Gaussian Splatting [15]. We optimize the splats in the local space by minimizing color loss on the rendering. We add and remove splats adaptively with their binding relation to triangles inherited so that all splats remain rigged throughout the optimization procedure. Further, we regularize the position and scaling of 3D Gaussian splats to suppress artifacts during animation.

use a photometric head tracker based on [39] to fit FLAME [21] parameters with multi-view observations and known camera parameters. The FLAME meshes have vertices at varied positions but share the same topology. Therefore, we can build a consistent connection between triangles of the mesh and 3D Gaussian splats (Sec. 3.2). The splats are rendered into images via a differentiable tile rasterizer [15]. These images are then supervised by the ground-truth images towards learning a photorealistic human head avatar. As per static scenes, it is also necessary to densify and prune Gaussian splats for optimal quality, with a set of adaptive density control operations [15]. To achieve this without breaking the connection between triangles and splats, we design a binding inheritance strategy (Sec. 3.3) so that new Gaussian points remain rigged to the FLAME mesh. In addition to the color loss, we also find it crucial to regularize the local position and scaling of Gaussian splats to avoid quality degradation under novel expressions and poses (Sec. 3.4).

3.1. Preliminary

3D Gaussian Splatting [15] provides a solution to reconstruct a static scene with anisotropic 3D Gaussians given images and camera parameters. A scene is represented by a set of Gaussian splats, each defined by a covariance matrix Σ centered at point (mean) $\boldsymbol{\mu}$:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \quad (1)$$

Note that covariance matrices have physical meaning only when they are semi-definite, which cannot be guaranteed for an optimization process with gradient descent. Therefore, Kerbl et al. [15] first define a parametric ellipse with a scaling matrix S and a rotation matrix R, then construct the covariance matrix by:

$$\Sigma = RSS^T R^T. \quad (2)$$

Practically, an ellipse is stored as a position vector $\boldsymbol{\mu} \in \mathbb{R}^3$, a scaling vector $\boldsymbol{s} \in \mathbb{R}^3$, and a quaternion $\boldsymbol{q} \in \mathbb{R}^4$. In our paper, we use $\boldsymbol{r} \in \mathbb{R}^{3 \times 3}$ to notate the corresponding rotation matrix to $\boldsymbol{q}$.

For rendering, the color $\boldsymbol{C}$ of a pixel is computed by blending all 3D Gaussians overlapping the pixel:

$$\boldsymbol{C} = \sum_{i=1} \boldsymbol{c}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j), \quad (3)$$

where $\boldsymbol{c}_i$ is the color of each point modeled by 3-degree spherical harmonics. The blending weight α' is given by

evaluating the 2D projection of the 3D Gaussian multiplied by a per-point opacity α. The Gaussian splats are sorted by depth before blending to respect visibility order.

3.2. 3D Gaussian Rigging

The key component of our method is how we build connections between the FLAME [21] mesh and 3D Gaussian splats. Initially, we pair each triangle of the mesh with a 3D Gaussian and let the 3D Gaussian move with the triangle across time steps. In other words, the 3D Gaussian is static in the local space of its parent triangle but dynamic in the global (metric) space as the triangle moves. Given the vertices and edges of a triangle, we take the mean position $\boldsymbol{T}$ of the vertices as the origin of the local space. Then, we concatenate the direction vector of one of the edges, the normal vector of the triangle, and their cross product as column vectors to form a rotation matrix $\boldsymbol{R}$, which describes the orientation of the triangle in the global space. We also compute a scalar k by the mean length of one of the edges and its perpendicular to describe the triangle scaling.

For the paired 3D Gaussian of a triangle, we define its location $\boldsymbol{\mu}$, rotation $\boldsymbol{r}$, and anisotropic scaling $\boldsymbol{s}$ all in the local space. We initialize the location $\boldsymbol{\mu}$ at the local origin, the rotation $\boldsymbol{r}$ as an identity rotation matrix, and the scaling $\boldsymbol{s}$ as a unit vector. At rendering time, we convert these properties into the global space by:

$$r' = Rr, \tag{4}$$
$$\mu' = kR\mu + T, \tag{5}$$
$$s' = ks. \tag{6}$$

We incorporate triangle scaling in Eqs. (5) and (6) so that the local position and scaling of a 3D Gaussian are defined relative to the absolute scale of a triangle. This enables an adaptive step size in the metric space with a constant learning rate for parameters defined in the local space. For example, a 3D Gaussian paired with a smaller triangle will move slower in an iteration step than those paired with large triangles. This also makes interpreting the parameters regarding the distance from the triangle center easier.

3.3. Binding Inheritance

Only having the same numbers of Gaussian splats as the triangles is insufficient to capture details. For instance, representing a curved hair strand requires multiple splats, while a triangle on the scalp may intersect with several strands. Therefore, we also need the adaptive density control strategy [15], which adds and removes splats based on the view-space positional gradient and the opacity of each Gaussian.

For each 3D Gaussian with a large view-space positional gradient, we split it into two smaller ones if it is large or clone it if it is small. We conduct this in the local space and ensure a newly created Gaussian is close to the old one that

triggers this densification operation. Then, it is reasonable to bind a new 3D Gaussian to the same triangle as the old one because it was created to enhance the fidelity of the local region. Therefore, each 3D Gaussian must carry one more parameter, the index of its parent triangle, to enable binding inheritance during densification.

Besides densification, we also use the pruning operation as a part of the adaptive density control strategy [15]. It periodically resets the opacity of all splats close to zero and removes points with opacity below a threshold. This technique is effective in suppressing floating artifacts, however, such pruning can also cause problems in a dynamic scene. For instance, regions of the face that are often occluded (such as eyeball triangles), can be overly sensitive to this pruning strategy, and often end up with few or no attached Gaussians. To prevent this, we keep track of the number of splats attached to each triangle, and ensure that every triangle always has at least one splat attached.

3.4. Optimization and Regularization

We supervise the rendered images with a combination of $\mathcal{L}_1$ term and a D-SSIM term following [15]:

$$\mathcal{L}_{\text{rgb}} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}}, \tag{7}$$

with $\lambda = 0.2$. This already results in good re-rendering quality without additional supervision, such as depth or silhouette supervision, thanks to the powerful tile rasterizer [15]. We found however, that if we try to animate these splats via FLAME [21] to novel expressions and poses, large spike- and blob-like artifacts appear wildly throughout the scene. This is due to a poor alignment between the Gaussian splats and the triangles.

Position loss with threshold. A basic hypothesis behind 3D Gaussian rigging is that the Gaussian splats should roughly match the underlying mesh. They should also match their locations; for instance a Gaussian representing a spot on the nose should not be rigged to a triangle on the cheek. Although our splats are initialized at triangle centers, and new splats are added nearby these existing ones, it is not guaranteed that the primitives remain close to their parent triangle after the optimization. To address this, we regularize the local *position* of each Gaussian by:

$$\mathcal{L}_{\text{position}} = \| \max(\boldsymbol{\mu}, \epsilon_{\text{position}}) \|_2, \tag{8}$$

where $\epsilon_{\text{position}} = 1$ is a threshold that tolerates small errors within the scaling of its parent triangle.

Scaling loss with threshold. Aside from position, the scaling of 3D Gaussians is even more essential for the visual quality during animation. Specifically, if a 3D Gaussian is large in comparison to its parent triangle, small rotations of the triangle – barely noticeable at the scale of the triangle – will be magnified by the scale of the 3D Gaussian, resulting

in unpleasant jittering artifacts. To mitigate this, we also regularize the local *scale* of each 3D Gaussian by:

$$\mathcal{L}_{\text{scaling}} = \| \max\left(\boldsymbol{s}, \epsilon_{\text{scaling}} \right) \|_2, \tag{9}$$

where $\epsilon_{\text{scaling}} = 0.6$ is a threshold that disables this loss term when the scale of a Gaussian less than $0.6\times$ the scale of its parent triangle. This $\epsilon_{\text{scaling}}$-tolerance is indispensable; without it, the Gaussian splats shrink excessively, causing rendering speeds to deteriorate, as camera rays need to hit more splats before zero transmittance is reached.

Our final loss function is thus:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{position}}\mathcal{L}_{\text{position}} + \lambda_{\text{scaling}}\mathcal{L}_{\text{scaling}}, \tag{10}$$

where $\lambda_{\text{position}} = 0.01$ and $\lambda_{\text{scaling}} = 1$. Note that we only apply $\mathcal{L}_{\text{position}}$ and $\mathcal{L}_{\text{scaling}}$ to visible splats. Thereby, we only regularize points when the color loss $\mathcal{L}_{\text{rgb}}$ is present. This helps to maintain the learned structure of often-occluded regions such as teeth and eyeballs.

Implementation details. We use Adam [17] for parameter optimization (the same hyperparameter values are used across all subjects). We set the learning rate to 5e-3 for the position and 1.7e-2 for the scaling of 3D Gaussians and keep the same learning rates as 3D Gaussian Splatting [15] for the rest of the parameters. Alongside the Gaussian splat parameters, we also finetune the translation, joint rotation, and expression parameters of FLAME [21] for each timestep, using learning rates 1e-6, 1e-5, and 1e-3, respectively. We train for 600,000 iterations, and exponentially decay the learning rate for the splat positions until the final iteration, where it reaches $0.01\times$ the initial value. We enable adaptive density control with binding inheritance every 2,000 iterations, from iteration 10,000 until the end. Every 60,000 iterations, we reset the Gaussians' opacities. We use a photo-metric head tracker to obtain the FLAME parameters, including shape β, translation $\boldsymbol{t}$, pose $\boldsymbol{\theta}$, expression ψ, and vertex offset $\Delta\boldsymbol{v}$ in the canonical space.

4. Experiments

4.1. Setup

Settings. We evaluate the quality of head avatars with three settings: 1) *novel-view synthesis*: driving an avatar with head poses and expressions from training sequences and rendering from a held-out viewpoint. 2) *self-reenactment*: driving an avatar with unseen poses and expressions from a held-out sequence of the same subject and rendering all 16 camera views. 3) *cross-identity reenactment*: animating an avatar with poses and expressions from another subject.
Dataset. We conduct experiments on video recordings of 9 subjects from the NeRSemble [18] dataset. All recordings contain 16 views covering the front and sides of a subject. For each subject, we take 11 video sequences and down-sample images to a resolution of 802×550. Participants

	Novel-View Synthesis			Self-Reenactment		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AvatarMAV [44]	29.5	0.913	0.152	24.3	0.887	0.168
PointAvatar [50]	25.8	0.893	0.097	23.4	0.884	0.102
INSTA [52]	26.7	0.899	0.122	26.3	0.906	0.110
Ours	31.6	0.938	0.065	26.0	0.910	0.076

Table 1. Quantitative comparison with state-of-the-art methods. Green indicates the best and yellow indicates the second.

were instructed to perform specific expressions or emotions in 10 sequences and freely perform in the last one. To conduct quantitative evaluation for each subject, we train our method and three baselines (INSTA [52], PointAvatar [50], and AvatarMAV [44]) using 9 out of the 10 prescribed sequences and 15 out of 16 available cameras. We use the 11th (free-performance) sequence to visually assess cross-identity reenactment capabilities. Please refer to the supplementary material for details of the dataset and baselines.

4.2. Head Avatar Reconstruction and Animation

We evaluate the reconstruction quality of avatars by novel-view synthesis and animation fidelity by self-reenactment. Fig. 3 shows qualitative comparisons. For novel-view synthesis, all methods produce plausible rendering results. A close inspection of PointAvatar's [50] results show dotted artifacts, owing its fixed point size. In our case, the anisotropic scaling of 3D Gaussians alleviates this issue.

INSTA [52] shows clean results for the face, however regions around the neck and shoulder can be noisy, as the tracked FLAME meshes are often misaligned in those regions. This causes issues for INSTA, as its warping process is based on the nearest triangle. In the case of our method, each Gaussian splat is rigged to a consistent triangle, regardless of the pose or expression. When the tracked mesh is inaccurate, the positional gradient to a Gaussian splat can consistently back-propagate to the same triangle. This enables misalignments due to incorrect FLAME tracking to be corrected while the 3D Gaussians are being optimized.

AvatarMAV [44] shows comparable quality to other methods in novel-view synthesis but struggles with novel-expression synthesis. This is because it only uses the expression vector of a 3DMM as conditioning. Since the control from the expression vectors to the deformation bases must be learned, it struggles to reproduce expressions that are far from the training distribution. Similar conclusions can be drawn from the quantitative comparison shown in Tab. 1. Our approach outperforms others by a large margin regarding metrics for novel-view synthesis. Our method also stands out in self-reenactment, with significantly lower perceptual differences in terms of LPIPS [48]. Note that self-reenactment is based on tracked FLAME meshes that

Figure 3. Qualitative comparison on novel-view synthesis and self-reenactment of head avatars. Our method outperforms state-of-the-art methods by producing significantly sharper rendering outputs. We obtain precise reconstruction of details such as reflective light on eyes, hair strands, teeth, etc. Our results for self-reenactment show more accurate expressions compare to baselines.
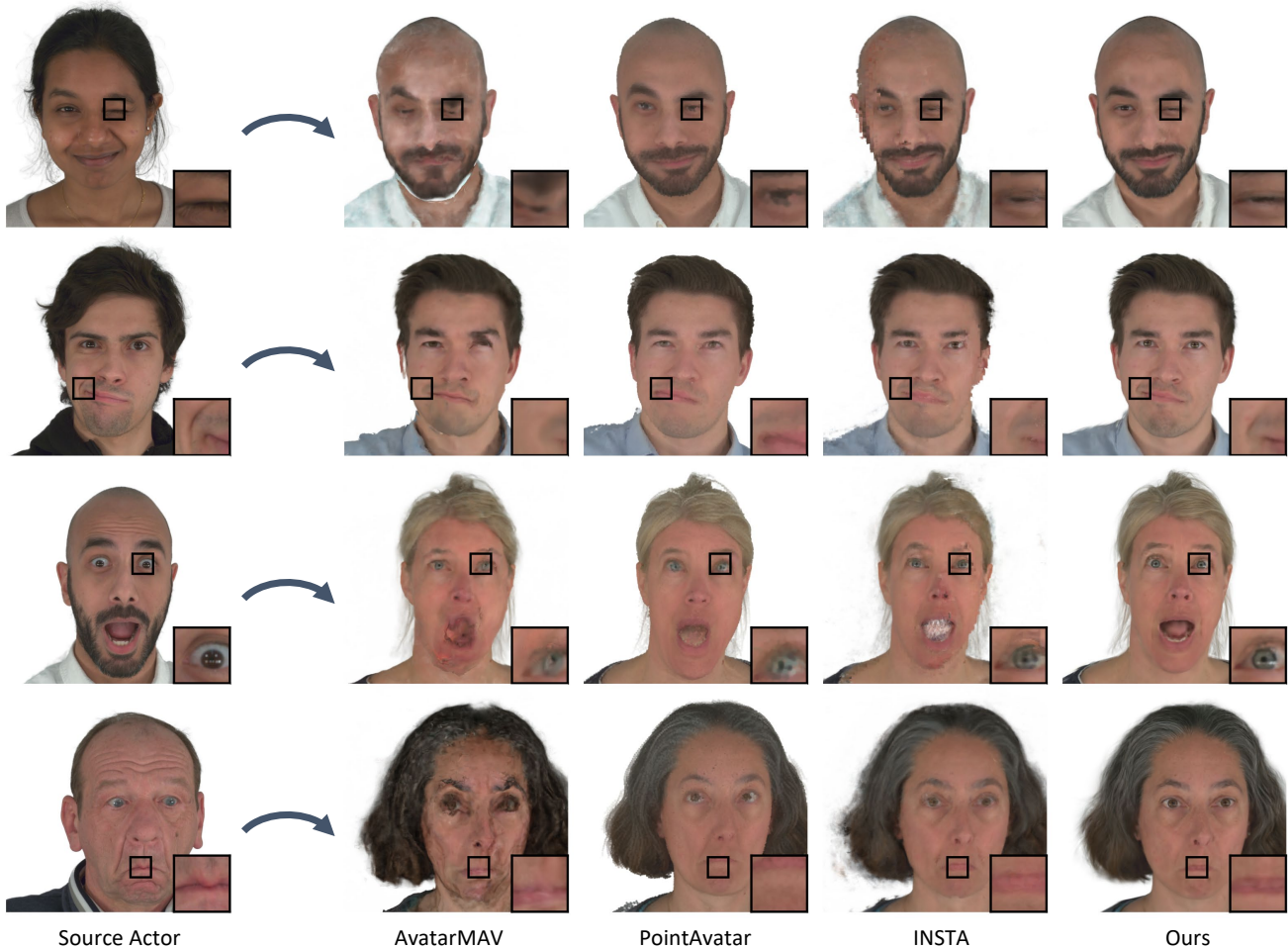
Figure 4. Cross-identity reenactment of head avatars. We use the tracked FLAME expression and pose parameters of source actors to drive the reconstructed avatars. Our method produces high-quality rendering and transfers expressions vividly, while baseline methods suffer from artifacts and generalize poorly to novel expressions.

may not perfectly align with the target images, thus bringing disadvantages to our results with more visual details regarding pixel-wise metrics such as PSNR.

For a real-world test of avatar animation, we conduct experiments on cross-identity reenactment in Fig. 4. Our avatars accurately reproduce eye blinks and mouth movements from source actors showing lively, complex dynamics such as wrinkles. INSTA [52] suffers from aliasing artifacts when the avatars move beyond the occupancy grid of I-NGP [29] optimized for training sequences. The movement of results from PointAvatar [50] is not precise because its deformation space is not guaranteed to be consistent with FLAME. AvatarMAV [44] exhibits large degradations in reenactment due to a lack of deformation priors.

4.3. Ablation Study

To validate the effectiveness of our method components, we deactivate each of them and report results in Tab. 2.
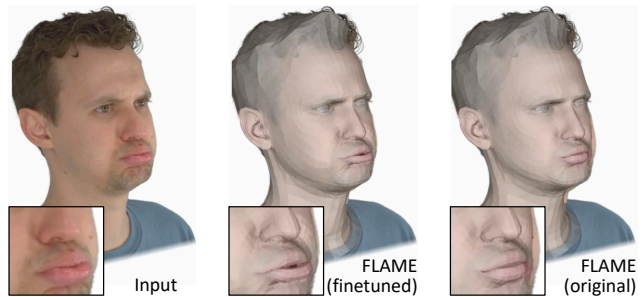

Figure 5. Fine-tuning FLAME parameters leads to better alignment of the mesh to the input image. In the example above, the movements of cheeks and lips are better captured after fine-tuning.

Adaptive density control with binding inheritance. Without binding inheritance, we lose the ability to add and remove Gaussian splats besides the initial ones. With a lim-

	Novel-View			Self-Reenactment		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Ours	28.8	0.883	0.098	25.1	0.853	0.101
w/o ADC	26.8	0.854	0.206	25.1	0.860	0.183
w/o $\mathcal{L}_{\text{scaling}}$	28.0	0.877	0.114	24.9	0.852	0.109
w/o $\epsilon_{\text{scaling}}$	25.0	0.833	0.195	24.1	0.843	0.176
w/o $\mathcal{L}_{\text{position}}$	29.7	0.894	0.091	24.9	0.851	0.096
w/o $\epsilon_{\text{position}}$	28.7	0.882	0.105	25.0	0.855	0.106
w/o FLAME ft.	26.1	0.855	0.131	25.5	0.862	0.124

Table 2. Ablation study on subject #304. Green indicates the best and yellow indicates the second. "ADC" refers to Adaptive Density Control with binding inheritance. 'FLAME ft.' refers to FLAME parameter fine-tuning.

Figure 6. The position loss and the scaling loss helps prevent artifacts during animation with novel expressions and poses.

ited number of splats, the scaling of each splat increases to occupy the same space, causing blurry renderings, leading to huge fidelity loss, as shown in the second row of Tab. 2.

Regularization on the scaling of Gaussian splats. Without the scaling loss, artifacts show up as spikes (the middle column in Fig. 6), and slightly worsen the image quality (the third row in Tab. 2). But when we use the scaling loss without an error tolerance, all metrics drastically deteriorate (the fourth row in Tab. 2) because all the splats are regularized to be infinitely small, making it hard to construct opaque surfaces.

Regularization on the position of Gaussian splats. When disabling the threshold for the position loss, we get slightly worse metrics (the sixth row in Tab. 2). But when turning off the position loss, the metrics on novel-view synthesis become the best in the table (the fifth row in Tab. 2). This is reasonable because without constraints to the mesh Gaussian splats can freely move in the space to achieve minimal re-rendering error. However, since the distribution of Gaussian splats are overfitted to training frames, the avatar shows artifacts such as cracks and fly-around blobs (the left column in Fig. 6) in novel expressions and poses. Therefore, the position loss is necessary to enforce a conservative deviation of splats from the mesh surface so that we can animate the reconstructed avatar with unseen facial motion.

FLAME fine-tuning. Thanks to our consistent binding between Gaussian splats and triangles, we can effectively fine-tune FLAME parameters while optimizing Gaussian splats (Fig. 5). As shown in the last row in Tab. 2, turning off the fine-tuning negatively affects image quality for novel-view synthesis. For self-reenactment, fine-tuning FLAME parameters also leads to lower perceptual error.

5. Limitations and Potential Negative Impacts

Our approach, based on 3D Gaussian splats, directly captures the radiance field without decoupling material and lighting. Hence, relighting the avatar is not feasible within
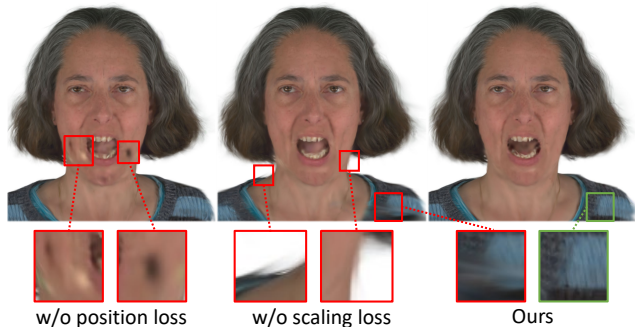
our current approach. Additionally, we lack control over areas that are not modeled by FLAME such as hair and other accessories. Here, we believe that modeling these parts directly would provide an interesting future research avenue; for instance, recent methods show promising result in reconstructing hair [35, 41].

The creation of photorealistic avatars may introduce a wide spectrum of malevolent uses. Privacy violations are a significant concern, with potential for unauthorized manipulation of people's likeness. Moreover the creation of deceptive content, like deepfake videos, can lead to misinformation, defamation and harm to reputations. Identity theft is an additional risk, as avatars may be exploited for impersonation and fraudulent activities. We strongly condemn unauthorized and malevolent uses of this technology, emphasizing the need for ethical considerations in all applications utilizing our method.

6. Conclusion

GaussianAvatars is a novel approach that creates photorealistic avatars from video sequences. It features a dynamic 3D representation based on 3D Gaussian splats that are rigged to a parametric morphable face model. This enables flexible control and precise attribute transfer of a dynamic, photorealistic avatar. The set of Gaussian splats can deviate from the mesh surface to compensate the absence or inaccuracy of the morphable model, exhibiting astonishing ability to model fine details on human heads. Our approach outperforms state-of-the-art methods on image quality and expression accuracy for a large margin, indicating a large potential for more applications in related domains.

Acknowledgements

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16610–16620, 2023. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1

[3] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003. 2

[4] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[5] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5933–5942, 2019. 2

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022. 1, 2

[7] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 2

[8] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2021. 2

[9] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 2

[10] 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 (Proceedings of SIGGRAPH Asia)*, 41(6), 2022. 2

[11] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18653–18664, 2022. 2

[12] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20374–20384, 2022. 2

[13] 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. 1

[14] 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. 2

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

[16] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM transactions on graphics (TOG)*, 37(4): 1–14, 2018. 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 42(4), 2023. 1, 5

[19] Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. Hugs: Human gaussian splats. *arXiv preprint arXiv:2311.17910*, 2023. 2

[20] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. *arXiv preprint arXiv:2311.16099*, 2023. 2

[21] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2, 3, 4, 5

[22] 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. 1

[23] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. *arXiv*, 2023. 2

[24] 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

[25] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 2

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

[27] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 2

[28] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1, 2, 7

[30] 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. 1, 2

[31] 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. *arXiv preprint arXiv:2106.13228*, 2021. 1, 2

[32] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009. 2

[33] 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

[34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[35] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In *European Conference on Computer Vision*, pages 73–89. Springer, 2022. 8

[36] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 2

[37] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 2

[38] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2

[39] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016. 2, 3

[40] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[41] Ziyan Wang, Giljoo Nam, Tuur Stuyck, Stephen Lombardi, Chen Cao, Jason Saragih, Michael Zollhöfer, Jessica Hodgins, and Christoph Lassner. Neuwigs: A neural dynamic model for volumetric hair capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8641–8651, 2023. 8

[42] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 2

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

[44] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2, 5, 7

[45] Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Huang Han, Qi Guojun, and Yebin Liu. Latentavatar: Learning latent expression code for expressive neural head avatar. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 2

[46] Keyang Ye, Tianjia Shao, and Kun Zhou. Animatable 3d gaussians for high-fidelity synthesis of human motions. *arXiv preprint arXiv:2311.13404*, 2023. 2

[47] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 2

[48] 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

[49] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13545–13555, 2022. 2

[50] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-

based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21057–21067, 2023. 2, 5, 7

[51] Wojciech Zielonka, Timur Bagautdinov, Shunsuke Saito, Michael Zollhöfer, Justus Thies, and Javier Romero. Drivable 3d gaussian avatars. 2023. 2

[52] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4574–4584, 2023. 2, 5, 7