Rig3DGS: Creating Controllable Portraits from Casual Monocular Videos

Alfredo Rivero*
Stony Brook University
arivero@cs.stonybrook.edu

ShahRukh Athar*
Stony Brook University, Captions
sathar@cs.stonybrook.edu

Zhixin Shu
Adobe Research
zshu@adobe.com

Dimitris Samaras
Stony Brook University
samaras@cs.stonybrook.edu

Figure 1. **Rig3DGS.** Our method, Rig3DGS, enables the creation of reanimatable portraits with full control over facial expressions, head-pose of a subject and the viewing direction of the entire scene they're in. Rig3DGS uses a deformation field that is constrained to lie in a subspace defined by the mesh deformation to ensure photorealistic reanimation and generalization to novel expressions and head-poses.

Abstract

*We present Rig3DGS, a novel technique for creating reanimatable 3D portraits from short monocular smartphone videos. Rig3DGS learns to reconstruct a set of controllable 3D Gaussians from a **monocular** video of a dynamic subject captured with varying head poses and facial expressions in an in-the-wild scene. In contrast to synchronized multi-view studio captures, this in-the-wild, single camera setup brings fresh challenges to learning high quality 3D Gaussians. We address these challenges by learning to deform 3D Gaussians from a fixed canonical space to the deformed space that is consistent with the target facial expression and head-pose. Our key contribution is a carefully designed deformation model that is guided by a 3D face morphable model. This deformation not only enables control over facial expression and head-poses but also allows our method to generates high-quality photorealistic renders of the whole scene. Once trained, Rig3DGS is able to generate photorealistic renders of a subject and their scene for novel facial expression, head-poses, and viewing directions. Through extensive experiments we demonstrate that Rig3DGS significantly outperforms prior art while being orders of magnitude faster.*

1. Introduction

Creating controllable 3D human portraits is crucial for various immersive experiences, including virtual reality, telepresence, film production, and educational applications. Yet, the realization of this technology for everyday consumers using only basic smartphone cameras presents considerable challenges. Modeling a 3D controllable portrait from videos typically involves either an explicit or implicit registration of a dynamic human subject[1] and accounting for varying factors such as facial expressions and head poses in each frame. This process requires a precise estimation of the facial deformations caused by these factors, which is often challenging without ground truth supervision. Modeling challenges are further compounded when using a monocular capture, as each head pose and expression is only visible

*Equal contribution

[1]Often via learning a deformation

from a single viewpoint making accurate estimation even harder.

While some prior work, such as RigNeRF [2], addresses these challenges, training and rendering are often very slow due to the use of a multi-layer perceptron (MLP) based neural radiance field (NeRF [26]) when modeling a 3D scene. The insufficient representation capacity of these MLPs, even with the use of a positional encoding, leads to blurry results, especially for novel expressions and poses. More recent 3D representations, such as 3D gaussians [21], significantly improve the rendering quality while being orders of magnitude faster to both train and render than MLP-based NeRFs [26]. However, current works on 3D Gaussian Splatting (3DGS) [21] cannot represent dynamic content and are unsuitable for reanimation tasks. An initial effort towards developing dynamic 3D gaussians [25] uses a multi-camera setup to reconstruct a point cloud of gaussians representing the dynamic scene at every time-step and establishes correspondences across gaussians when training. However, this method only reconstructs the dynamic scene and cannot be reanimated.

In a monocular setting, reconstructing a scene's point cloud in its dynamic regions is difficult [31] due to only a single view being available at each time-step. Consequently, [25]'s approach cannot establish weak correspondences across time in the dynamic regions of a scene during reconstruction. In this paper, we present Rig3DGS, a method that "rigs" 3D gaussians and enables the creation of reanimatable neural-gaussian portraits. Rig3DGS utilizes a point cloud of 3D gaussians in a canonical space, initialized using COLMAP [31] for static regions of a scene and a FLAME [23] mesh for dynamic regions involving the human subject of a scene. These canonical 3D gaussians are then transformed to a facial expression and head-pose dependent deformed space, where they are rendered via differentiable Gaussian Splatting. Our key insight is to restrict the deformation from the canonical to the deformed space to lie in the subspace defined using the vertices of a 3D face mesh. More specifically, we predict the deformation from the canonical space to deformed space for each 3D gaussian as a weighted sum of the deformations of its k-closest vertices on a morphable model mesh. These learnable weights are optimized using a photometric loss w.r.t to a ground truth image. By restricting each gaussian's deformation to lie in a linear subspace of the vertex deformations, we can effectively regularize the otherwise ill-posed problem of learning a per-point deformation without any ground truth or multi-view supervision available. In Sect 4.5, we show that this formulation is essential for better generalization to novel facial expressions and head-poses. Following training, Rig3DGS allows for the creation of reanimatable portraits, providing control over the facial expressions and head-pose of the subject, as well as enabling the synthesis of novel views of the entire scene. In summary, our contributions are as follows:

- We propose a novel deformation model learnt in the subspace of the deformation defined by a 3D morphable model. This enables us to generalize to novel facial expressions and head-poses during reanimation.
- We propose Rig3DGS, a method that rigs 3D gaussians to enable the creation of reanimatable portraits with full control over facial expressions, head-pose, and novel view synthesis of the entire scene from a casually-captured *monocular* smartphone video.
- We demonstrate significant improvements in rendering quality over prior work on neural portraits with novel facial expressions, head-poses, and novel views synthesis of the entire scene while being 100 times faster than prior work due to representing the scene using 3D gaussians.

2. Related Work

Rig3DGS is a method for arbitrary facial expression control and novel view synthesis of scenes captured in portrait videos. It is closely related to recent work on neural rendering and novel view synthesis, 3D facial modeling, and controllable face generation.

Neural Scene Representations and Novel View Synthesis. Rig3DGS is related to recent advances in neural rendering and novel view synthesis [3, 4, 10, 15, 21, 24–28, 30, 32, 37, 40]. Neural Radiance Fields (NeRFs [26]) learn a volumetric representation of a scene which typically, when provided a 3D point and the direction from which the point is being viewed, predicts color and volume density using a differentiable renderer. Our method is built upon 3D Gaussian Splatting [21] and uses a dense point cloud of three-dimensional gaussian kernels to represent a scene's geometry. For any given camera pose, 3D gaussian kernels are projected onto the image evaluated densely pixel-by-pixel. Gaussian kernels within the local neighborhood of each pixel are sorted and pixel color is approximated using a modified approach to conventional α-blending. 3D Gaussian Splatting [21] and other NeRFs [26] are trained by minimizing the error between the predicted color of a pixel and its ground truth value. While NeRFs [26] are able to generate high quality and photo-realistic images for novel view synthesis, many are designed for static scenes and are unable to represent scene dynamism.

Dynamic Neural Scene Representations. Methods such as [14, 24, 25, 30] extend NeRFs [26] to dynamic scenes through specialized frame-to-frame parameter residual learning [25] or incorporating a time component and canonical deformation network [15, 24, 30]. Among approaches incorporating a canonical deformation network, a

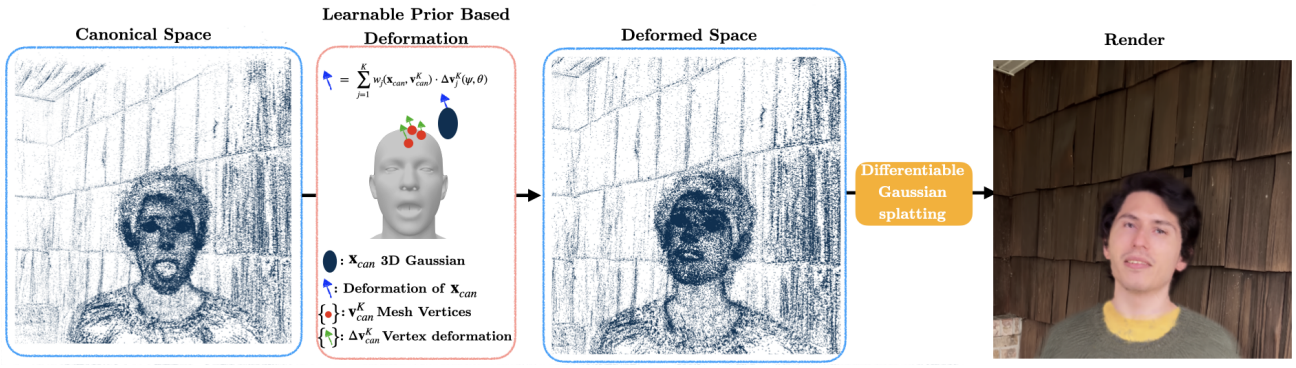

Figure 2. **Rig3DGS.** Our method models the dynamic scene as a collection of 3D gaussians in a canonical space that are deformed according to a target facial expression and head-pose to the deformed space before being rendered via differentiable splatting. We constrain the deformation to lie in the subspace of local vertex deformations, which allows us to generate photorealistic renders with high fidelity to target expressions and head-poses.

dynamic scene is decomposed into canonical and deformed spaces, with each moment in time defining a reconfiguration of the canonical space using learned displacements in position, color, and other parameters. Recent approaches to human body and facial reconstruction [2, 32, 37, 38, 40] refine and expand existing deformation-based NeRFs [26] for their data domains, aiming to reconstruct and reanimate data with the assistance of existing templates and/or statistical models. Among these domain-specialized approaches, FlashAvatar [37] and SplattingAvatar [32], both 3D Gaussian Splatting [21] avatar models, incorporates a 3D Morphable Model within their facial deformation network, influencing canonical gaussian displacement using an existing target facial mesh template.

Controllable Face Generation. Recent breakthroughs in Generative Adversarial Networks (GANs) [11, 13, 18–20, 39] have enabled high-quality 2D image generation and manipulation. These 2D breakthroughs have inspired a large collection of work [1, 5–7, 22, 29, 33–36] focusing on face image manipulation and editing. However, the majority of these works are intrinsically image-based and lack an explicit 3D representation. Therefore, these methods find it challenging to enable high-quality view synthesis and control portraits with large head pose changes or extreme facial expressions. Recent methods such as [32, 37, 40] address these shortcomings. Existing dynamic neural rendering models that use 3D Morphable Models such as FlashAvatar [37], SplattingAvatar [32], reconstruct and reanimate human heads at a high quality and with reasonable-to-quick rendering/training times. However, these methods fail to capture an entire portrait scene, with both of these methods failing to capture a scene's background. Geometrically-complete approaches such as RigNeRF [2] exist, which incorporates a deformation prior defined by an existing FLAME [23] mesh and corrected through residual learning. But, this method fails to reproduce facial portraits with reasonable rendering/training times.

3. Rig3DGS

In this section, we describe our method Rig3DGS, which enables novel view synthesis of neural portraits with arbitrary control over head pose and facial expressions. We represent a scene using 3D gaussians in the canonical space that are transformed to the deformed space using a constrained gaussian deformation model. More specifically, we constrain the deformation of a gaussian to lie in the subspace spanned by the deformation of its K-nearest vertices on the FLAME [23] mesh. Similarly, the rotation of each gaussian is learnt as a correction to the rotation of its K-nearest vertices. An overview of our method is given in Fig 2.

3.1. Preliminaries

3.1.1 3D Gaussian Splatting

3D Gaussian Splatting works by representing a scene using three-dimensional gaussians, each with a learnable spatial extent, orientation, opacity and color. More specifically, each gaussian is defined by a covariance matrix Σ and mean $\mathbf{x}$ as

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

where Σ is decomposed into two differentiable scaling and rotation matrices $\mathbf{S}$, $\mathbf{R}$ such that

$$\Sigma = \mathbf{RSS}^T\mathbf{R}^T.$$

When rendering novel views, each gaussian is projected onto the image plane using a differentiable variant of Surface Splatting [41]. Using the viewing transform matrix W

and the Jacobian matrix J of the affine approximation of the projective transformation, a 3D gaussian's covariance matrix Σ' in the camera co-ordinates is defined as

$$\Sigma' = JW\Sigma W^T J^T,$$

[41] uses the Σ' to define planar covariance which is then used for splatting. During splatting, all gaussians are sorted relative to their 3D mean's distance to the image plane, and a pixel's color C is computed as

$$C = \sum_{i=1}^{N} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j)$$

where c_i is the color of each point and α_i is given by evaluating a 2D gaussian with covariance Σ' multiplied with each gaussian's opacity.

3.2. Canonical/Deformed Space and Initialization

3D Gaussian Splatting, as described in Sect 3.1.1, does not support dynamic scene rendering, as the initial point cloud is assumed to be static. In contrast, our scene is inherently dynamic with the subject constantly assuming different facial expressions and head-poses. We choose to model this dynamic behaviour as a deformation from a predefined canonical space, where gaussians lie with positions $\mathbf{x}_{can}$, to a deformed space with gaussian positions $\mathbf{x}_{def}$. At render time, our scene is always rendered in deformed space.

Prior to optimization, we initialize the gaussians as

$$\mathbf{x}_{can}^{init} = \{\mathbf{x}_{can}^{COLMAP} \cup \mathbf{v}_{can}\} \tag{1}$$

which is a concatenation of background points given by COLMAP [31] $\mathbf{x}_{can}^{COLMAP}$ and the vertices of a canonical FLAME [23] mesh $\mathbf{v}_{can}$. We choose the canonical configuration of our fitted FLAME [23] mesh to be extracted from a close-to-neutral head pose frame within our data; we modify the jaw pose parameters θ_{jaw} of this mesh to have an open mouth. Empirically, we find an open mouth improves dynamic mouth modeling. Using [21]'s original gaussian densification scheme where $\mathbf{x}_{can}^{init}$ will be progressively pruned and densified into a set of gaussians with positions $\mathbf{x}_{can}$. However, our initial vertex gaussians with positions $\mathbf{v}_{can}$ are never pruned as they are integral to our deformation approach.

3.3. Deforming Gaussians with a 3DMM-Based Prior

Using expression and head-pose FLAME [23] parameters $\{\psi, \theta\}$ sourced from frame n, the deformation of our FLAME [23] vertices relative to their canonical configuration parameterized by $\{\beta_{can}, \psi_{can}, \theta_{can}\}$ can be defined as

$$\Delta\mathbf{v}(\psi,\theta) = M_{def} - M_{can} = M(\beta_{can},\psi,\theta) - M(\beta_{can},\theta_{can},\psi_{can}) \tag{2}$$

For each gaussian i with canonical position $\mathbf{x}_{i,can}$, we define its position in deformed space $\mathbf{x}_{i,def}$ using a 3DMM-based deformation D, a learned corrective transformation η, and minimal temporal deformation T

$$\mathbf{x}_{i,def} = \mathbf{x}_{i,can} + D + \eta + T(n) \tag{3}$$

3.3.1 3DMM-Based Deformation

Our 3DMM-based deformation D is a linear combination of the vertex deformation $\Delta\mathbf{v}(\psi,\theta)$ experienced by i's K-nearest mesh vertices $\mathbf{v}_{can}^K$

$$D(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K, \psi, \theta) = \textstyle\sum_{j=1}^{K} w_j(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K) \cdot \Delta\mathbf{v}_j^K(\psi,\theta) \tag{4}$$

Empirically, we find $K = 10$ produces reasonable results. The weights of our linear combination of vertices predicted using a 6-level multi-resolution tri-plane H and small two-layer MLP G as follows

$$w_j(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K) = G(H(\mathbf{x}_{i,can}), d_K(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K)) \tag{5}$$

3.3.2 Corrective and Temporal Deformation

Occasionally, our estimated deformation from $\mathbf{v}_{can}$ can be slightly misaligned with the displacement of a person's head in 3D space; η corrects for this scenario using a small two-layer MLP conditioned on both $\mathbf{x}_{i,can}$ and the average inverse-squared distance between $\mathbf{x}_{i,can}$ and $\mathbf{v}_{can}^K$, $d_K(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K)$, as follows:

$$\begin{gathered}\eta = \eta(\mathbf{x}_{i,can}, d_K(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K)) \\ \text{where} \quad d_K(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K) = \exp(\frac{1}{K}\sum_{j=1}^{K} |\mathbf{x}_{i,can} - \mathbf{v}_{j,can}^K|_2^2)^{-1}\end{gathered} \tag{6}$$

Our temporal deformation T accounts for any deformations not explainable by our 3DMM-based deformation (such as movement in the body) using a small two-layer MLP conditioned on both $\mathbf{x}_{i,can}$ and frame-index n.

$$T(n) = T(\mathbf{x}_{i,can}, n) \tag{7}$$

At render time, we set $n = 0$ and cache T's deformation for faster rendering. Empirically, we have found that magnitude of the deformation of η and $T(n)$ to be a factor of 10 or more smaller than that of D for large head-pose and expression articulations.

3.4. Rotating and Scaling Gaussians with a 3DMM-Based Prior

As described in Sect 3.1.1, each gaussian i we use to represent our scene in canonical space also has an associated rotation $\mathbf{R}_{i,can}$ and scale $\mathbf{S}_{i,can}$. Consequently, in order to correctly orient our gaussians in deformed space, we must

also predict an accompanying rotation and scale. We condition a pair of two-layer MLPs, R_ψ and S_ω, on $\mathbf{x}_{i,can}$ and d_K to predict both of these displacements. In the case of our rotation, we utilize the Kabsch [17] algorithm to calculate the rotation of i's K-nearest FLAME [23] vertices $\mathbf{v}_{can}^K$ in the canonical space. Thus, we define each gaussian i's accompanying rotation and scale in deformed space as

$$\begin{aligned} \mathbf{R}_{i,def} &= R_\psi(\mathbf{x}_{i,can}, d_K) \cdot R_{kabsch}(\mathbf{v}_{def}^K, \mathbf{v}_{can}^K) \cdot \mathbf{R}_{i,can} \\ \mathbf{S}_{i,def} &= S_\omega(\mathbf{x}_{i,can}, d_K) \cdot \mathbf{S}_{i,can} \end{aligned} \tag{8}$$

where, $R_{kabsch}(\mathbf{v}_{def}^K, \mathbf{v}_{can}^K)$ is the rotation estimate between $\mathbf{v}_{def}^K$ and $\mathbf{v}_{can}^K$ calculated using the Kabsch [17] algorithm.

3.5. Optimization

Given a video frame I, we supervise Rig3DGS using a photometric loss, FLAME [23] mesh vertex loss, and a variety of regularizations on our networks' outputs. Our final loss can be described fully as

$$\mathcal{L} = \mathcal{L}_{photo} + \mathcal{L}_{FLAME} + \mathcal{L}_{def} + \mathcal{L}_{reg} \tag{9}$$

3.5.1 Photometric Loss

We use a combination of an L1 loss and Structural Similarity (SSIM) score between I and Rig3DGS's render I_r

$$\mathcal{L}_{photo} = 0.8 \cdot |I - I_r|_1 + 0.2 \cdot SSIM(I, I_r) \tag{10}$$

3.5.2 FLAME Vertex Similarity Loss

We ensure that our 3DMM-based deformation D produces deformations similar to the vertex deformation $\Delta\mathbf{v}(\psi, \theta)$ defined by M_{def} and M_{can} for gaussians at mesh vertices $\mathbf{v}_{can}$

$$\mathcal{L}_{FLAME} = |D(\mathbf{v}_{can}, \mathbf{v}_{can}^K, \psi, \theta) - \Delta\mathbf{v}(\psi, \theta)|_2^2 \tag{11}$$

3.5.3 Deformation Distance Loss

We regularize our corrected deformation $D + \eta$ and T to be close to zero for points that are further than a pre-defined threshold δ from the mean center point of our mesh $\boldsymbol{x}_{M_{can}}^{center}$

$$\begin{aligned} &\mathcal{L}_{def} = \lambda_{def} \cdot |D(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K, \psi, \theta) + \eta(\mathbf{x}_{i,can}, d_K(\mathbf{x}_{i,can}, \mathbf{v}_{can}^K))|_2^2 \\ &\text{for } \mathbf{x}_{i,can} \text{ s.t.} \quad |\mathbf{x}_{i,can} - \boldsymbol{x}_{M_{can}}^{center}|_2 > \delta \end{aligned} \tag{12}$$

δ must be set per-subject, but can be reasonably approximated as 2 times the radius of a bounding sphere encompassing the modeled subject mesh M_{can}. Empirically, we find the ideal weight for $\lambda_{def} = 1e-1$.

3.5.4 Network Regularization Loss

In order to ensure a smooth optimization and prevent our deformation from learning local minima, we regularize the output of our networks R, S, η, and T.

$$\mathcal{L}_{reg} = \lambda_R|R|_2^2 + \lambda_S|1 - S|_2^2 + \lambda_\eta|\eta|_2^2 + \lambda_T|T|_2^2 \tag{13}$$

Empirically, we find the ideal weights as $\lambda_R = 1e{-}1$, $\lambda_S = 1.0$, $\lambda_\eta = 1e-3$, and $\lambda_T = 1e-3$.

3.5.5 Training

We train for 3,000 iterations without any deformation and with gaussian densification and pruning enabled. After this initial training period, we enable our deformation $D + \eta + T(n)$ and disable densification and pruning at 5,000 iterations. We train for a total of 60,000 iterations per portrait with a training time 5 hours on an NVIDIA RTX 3090. We utilize PyTorch's default Adam optimizer with a learning rate of 0.001.

4. Results

In this section, we show results of head-pose control, facial expression control, and novel view synthesis using Rig3DGS. For each scene, our model is trained on a short portrait video captured using a consumer smartphone.

4.1. Baseline Approaches

In the context of neural portrait reanimation, the only prior work that offers full control over facial expression, head-pose and viewing direction of the scene is RigNeRF [2]. RigNeRF [2] represents the dynamic scene as a neural radiance field where the mapping from the deformed space to the canonical space is performed using a 3DMM-based deformation. However, since RigNeRF [2] uses MLPs to model both the deformation and radiance field, it is extremely slow to train and render during inference. At an inference rate of about 18 FPS on a NVIDIA RTX 3090, we are about 100 times faster than RigNeRF [2]. The authors of RigNeRF [2] kindly provided results of their method on the training data used for this paper. A class of methods that are closely related to Rig3DGS are ones that only control facial expression and head-pose and *do not* model the entire scene. Prior works such as INSTA [40], FlashAvatar [37], and SplattingAvatar [32] create controllable human heads from monocular smartphone videos. INSTA [40] is built on Instant-NGP [27], a notably fast and high-quality [27] approach to neural rendering, and was previously state-of-the-art in terms of quality and speed for human head avatar models. FlashAvatar [37] and SplattingAvatar [41] are now state-of-the art human avatar models based on Gaussian Splatting, with both models providing valuable insights into different approaches to splatting-based facial modeling.

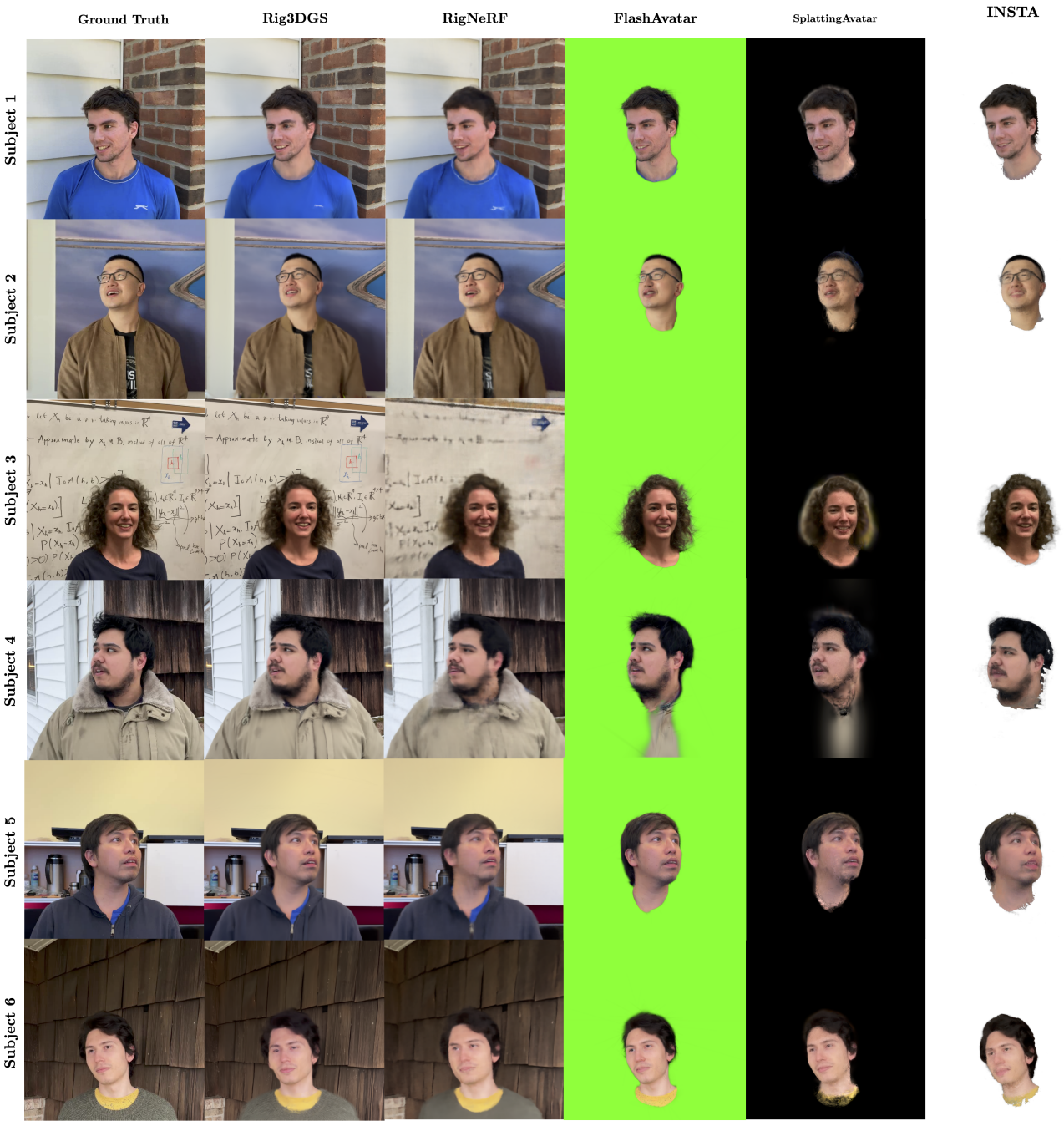

Figure 3. Qualitative comparison of Subjects 1-6 in Setting 1. Rig3DGS produces full-scene renders with high-quality facial and background detail that remains competitive with SOTA splatting-based avatar models.

4.2. Training Data Capture

We captured the training data using various iPhone models, including an iPhone XR, 12, and 13 Pro Max. The capture process is comprised of two distinct phases. In the first half, subjects are instructed to perform a diverse array of expressions and speech while maintaining a stationary head with the camera panning around them. Subsequently, in the latter half, the camera is fixed at head-level and subjects are prompted to rotate their heads enacting various facial expressions. All training videos are about 40-70 seconds long, equivalent to approximately 1200-2100 frames, and are down-sampled to a resolution of 512 x

	Rig3DGS (Ours)				*FlashAvatar [37]*				*SplattingAvatar [32]*				*RigNeRF [2]*				*INSTA [40]*			
Subject (Head Region)	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Subject 1	31.70	0.9591	0.0342	0.0845	29.54	0.9537	0.0265	0.0799	30.36	0.9455	0.0466	0.1037	30.19	0.9505	0.0469	0.1296	29.25	0.9530	0.0396	0.0986
Subject 2	33.10	0.9749	0.0239	0.0739	31.41	0.9759	0.0171	0.0862	27.97	0.9478	0.0455	0.1374	31.91	0.9715	0.0266	0.0841	30.74	0.9694	0.0337	0.1098
Subject 3	26.98	0.9240	0.0577	0.1242	28.57	0.9321	0.0379	0.0733	27.12	0.9226	0.0754	0.1670	26.35	0.9100	0.0607	0.1583	25.34	0.9185	0.0769	0.1718
Subject 4	30.23	0.9464	0.0529	0.1361	24.21	0.9175	0.0720	0.1403	27.72	0.9302	0.0758	0.1533	29.69	0.9400	0.0705	0.1771	28.29	0.9433	0.0615	0.1361
Subject 5	27.49	0.9668	0.0352	0.0797	28.66	0.9544	0.0328	0.0904	29.48	0.9502	0.0623	0.1330	27.23	0.9605	0.0489	0.1233	28.58	0.9559	0.0436	0.1011
Subject 6	24.07	0.9248	0.0623	0.1250	26.76	0.9452	0.0474	0.1107	26.21	0.9351	0.0639	0.1422	23.20	0.9232	0.0688	0.1444	23.60	0.9414	0.0619	0.1403
Average	**28.93**	**0.9493**	**0.0444**	**0.1039**	**28.19**	**0.9465**	**0.0389**	**0.0968**	**28.65**	**0.9409**	**0.0575**	**0.1330**	**28.10**	**0.9426**	**0.0537**	**0.1361**	**27.63**	**0.9469**	**0.0529**	**0.1263**

Table 1. Quantitative results for Subject 1-6's masked head region in Setting 1. Here we calculate PSNR, SSIM, LPIPS, and DISTS using head masks provided by INSTA. Our results are better than RigNeRF [2], INSTA [40] and SplattingAvatar [32] in the vast majority of metrics and remains competitive with FlashAvatar [37]. Best metrics are labeled in *pink* and second best in *yellow*.

	Rig3DGS (Ours)				*RigNeRF [2]*			
Subject (Full Scene)	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Subject 1	26.89	0.8684	0.1583	0.1088	25.04	0.8083	0.2359	0.1543
Subject 2	25.84	0.8665	0.1556	0.1226	23.31	0.7922	0.1779	0.1297
Subject 3	23.97	0.8670	0.1174	0.0747	20.34	0.6627	0.4264	0.2955
Subject 4	23.65	0.7910	0.1723	0.0960	22.53	0.6770	0.3314	0.1982
Subject 5	29.94	0.9304	0.0874	0.0674	28.71	0.9081	0.1266	0.1347
Subject 6	22.36	0.7503	0.1758	0.1246	21.22	0.6679	0.3826	0.2379

Table 2. Quantitative results for Subject 1-6's full-scene renders from Rig3DGS and RigNeRF [2] in Setting 1.

	Rig3DGS (Ours)				*RigNeRF [2]*			
Subject (Full Scene)	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Subject 1	28.05	0.8805	0.1394	0.0798	26.38	0.8203	0.2382	0.1491
Subject 2	25.94	0.8720	0.1729	0.1162	22.23	0.7900	0.2375	0.1471
Subject 3	26.38	0.8814	0.1025	0.0591	22.30	0.7096	0.4086	0.2840
Subject 4	25.92	0.8467	0.1419	0.0925	24.12	0.7468	0.2774	0.1790
Subject 5	28.58	0.9195	0.0961	0.0664	25.64	0.8269	0.2126	0.1656
Subject 6	27.51	0.8169	0.1635	0.1171	25.49	0.7339	0.3054	0.1968

Table 3. Quantitative results for Subject 1-6's full-scene renders from Rig3DGS and RigNeRF [2] in Setting 2.

512. We use COLMAP [31] to estimate camera parameters and Rig3DGS's inital point estimate in the static regions of the scene. In order for camera calibration to be accurate, we mask out the dynamic foreground prior to running COLMAP [31]. All models utilize FLAME [23] as their 3DMM of choice. For Rig3DGS we used DECA [9] to calculate an initial estimate of FLAME [23] parameters for each video, which we then optimize through standard landmark fitting, using landmarks predicted by 3DDFA-V2 [12] and camera parameters estimated from COLMAP [31].

4.3. Evaluation on Test Data

We assess the performance of Rig3DGS, RigNeRF [2], FlashAvatar [37], SplattingAvatar [41], and INSTA [40] using held-out images extracted from captured video sequences. We test two different settings: 1) **Fixed Camera view, changing expression and head-pose**: We fix the camera view and only vary expressions and head-poses to evaluate their fidelity during reanimation. Metrics for all methods are calculated on the head-region only. 2) **Fixed Head-pose with changing expression and camera view**: We fix the head-pose and only vary the camera view and expressions to evaluate fidelity of view synthesis and facial expressions. In this setting, we only compare Rig3DGS and RigNeRF, since other methods do not model the entire scene. We use about 30-60 *held-out* images for both settings. In both settings, we measure PSNR, SSIM, LPIPS [16], and DISTS [8]. While PSNR and SSIM measure pixel-wise accuracy, LPIPS [16] and DISTS [8] measure perceptual accuracy.

4.3.1 Evaluation in Setting 1

In Table 1 we show quantitative results of the first setting by measuring the metrics only on the head-region with constant view but changing expression and head-pose. As can be seen, across most subjects and metrics, Rig3DGS outperforms prior work by a large margin. This can also be seen in our qualitative results in Fig 3, where Rig3DGS generates higher quality renders than prior work with higher fidelity facial expressions and head-poses. While RigNeRF [2] is able to reasonably reproduce different head-poses and facial expressions, its renders are blurry and lack detail, especially around the eyes and mouth regions. This can be seen most clearly in the case of Subjects 1,3,4,5 and 6. In some cases, like that of Subject 4, RigNeRF [2] fails to reconstruct the background correctly. We attribute this to its use of MLPs that often underfit scenes. Please note, in the original paper [2], the authors train RigNeRF [2] on a resolution of 256x256 while for our paper, they trained it on a resolution of 512x512. In Table 2, we quantitatively evaluate the performance of Rig3DGS and RigNeRF [2] on the full scene. We omit INSTA [40] and PointAvatar since they do not model the full scene. As can be seen, Rig3DGS outperforms RigNeRF [2] across almost all metrics and on all subjects except for the PSNR of Subject 3. Similar to RigNeRF [2], while INSTA [40] is able to generate renders that are reasonably consistent with articulated facial expression and head-pose, it generates artifacts around the mouth, as can be seen in Subjects 2,3,5 and 6. Artifacts are also present in the hair region (Subjects 2,4,5 and 7) and on accesories such as glasses in the case of Subject 2. Unlike Rig3DGS, INSTA [40] is unable to model the full scene. Similar to INSTA [40], FlashAvatar [37] and SplattingAvatar [41] generate significant artifacts around image boundary and mouth region. In contrast to prior work, Rig3DGS is able to model facial expression and head-pose at a high fidelity to generate high-quality photorealistic renders.

	Subspace constrained (Ours)				*Fixed Prior*				*No Prior*			
Subject (Head Region)	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Subject 1	31.70	0.9591	0.0342	0.0845	30.57	0.9500	0.04152	0.0967	19.25	0.910	0.0868	0.224
Subject 5	30.23	0.9464	0.0529	0.1361	29.81	0.9444	0.0613	0.1513	20.78	0.896	0.09	0.199

Table 4. Quantitative ablative study on the efficacy of a subspace constrained deformation model. We narrow our study to Subjects 1 and 5.

4.3.2 Evaluation in Setting 2

In this setting, we evaluate the ability of Rig3DGS and RigNeRF [2] to model facial expressions and the full 3D scene. As can be seen in Table 3, Rig3DGS outperforms RigNeRF across all metrics and all subjects. This is due to RigNeRF's [2] renders being generally blurry compared to those of Rig3DGS. In our supplementary material, we include more qualitative results to support this claim.

4.4. Reanimation with Pose and Expression Control

In Fig 5, we show results of Rig3DGS being driven by a driving frame in 3 different novel views. As can be seen, across all subjects and driving frames, Rig3DGS is able to reproduce the driving expression and head-pose with high fidelity while simultaneously generating a faithful of the full scene under novel views. We strongly urge the reader to check out the supplementary material for more results on 3DMM driven reanimation and novel view synthesis.

4.5. Ablating the Deformation Model

In this section, we ablate our 3DMM-based deformation as defined in Eq. (4) and Eq. (5). We ablate the following 3 deformation models

- No Prior: Our deformation D is directly predicted by an MLP as follows

$$D = D(\mathbf{x}_{can}, \psi, \theta) = \text{MLP}(\mathbf{x}_{can}, \psi, \theta) \qquad (14)$$

- Deformation with Fixed Prior: Our deformation D uses only an exponentially-attenuated 3DMM-deformation (similar to RigNeRF[2])

$$w_j(\mathbf{x}_{i,can}, \mathbf{v}^K_{can}) = d_K(\mathbf{x}_{i,can}, \mathbf{v}^K_{can})$$

- Subspace constrained Deformation: Following Eq. (3)

Figure 4. Ablation of the Learnable Deformation Prior. As can be seen, the deformation, as defined by Eq. (3), is able to model target expressions and head-poses better than the fixed prior (see highlighted regions). The model with no prior fails to reanimate altogether.

In Fig 4, we show qualitative results of these three deformation approaches for a novel expression and head-pose. As can be seen, the model without a prior is unable to generate any meaningful render. The model with a fixed prior, similar to what RigNeRF [2] uses, is unable to model fine expressions and gives a somewhat blurry render. In contrast, the proposed subspace constrained deformation from Eq. (3) is able to generate a photorealistic render with high fidelity to the target facial expression and head-pose. In Table 4, we provide a quantitative comparison between the three deformation models. As can be seen, using our proposed deformation, as defined in Eq. (3) gives us the best results across all metrics on both subjects.

Figure 5. Sample renders of Subjects 1 and 4 reanimated using different expression and head pose donors. We refer the reader to our supplementary video material for a more comprehensive evaluation.

5. Conclusion and Limitations

In this paper we have presented Rig3DGS, a novel method capable of arbitrary facial expression control and novel view synthesis for portrait videos. Rig3DGS uses a learnable deformation prior to ensure stability during training and generalization to novel facial expressions, head-pose, and viewing direction. Rig3DGS is also able to model details of the subject's face such as hair and glasses and reproduce them with high fidelity as the video is driven.

However, challenges remain. Rig3DGS is unable to model strong non-uniform illumination and requires the subject in the portrait video to remain relatively still during capture. Additionally, Rig3DGS cannot model the interior of the mouth faithfully, such as the tougue and inner teeth. We hope to address this in future work.

References

[1] S Athar, Z Shu, and D Samaras. Self-supervised deformation modeling for facial expression editing. In *IEEE FG*, 2020.

[2] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

[4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022.

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

[7] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5154–5163, 2020.

[8] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *CoRR*, abs/2004.07728, 2020.

[9] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. 2021.

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

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[12] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[14] Hankyu Jang and Daeyoung Kim. D-tensorf: Tensorial radiance fields for dynamic scenes. *arXiv preprint arXiv:2212.02375*, 2022.

[15] Hankyu Jang and Daeyoung Kim. D-tensorf: Tensorial radiance fields for dynamic scenes, 2022.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[17] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

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

[22] Marek Kowalski, Stephan J Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 299–315. Springer, 2020.

[23] 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), 2017.

[24] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *arXiv preprint arXiv:2205.15723*, 2022.

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

[26] 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. 2020.

[27] 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.

[28] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and

Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[29] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: One-shot anatomically consistent facial animation. *International Journal of Computer Vision*, 128(3):698–713, 2020.

[30] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2021.

[31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] 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 (CVPR)*, 2024.

[33] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In *CVPR*, 2017.

[34] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *ECCV*, 2018.

[35] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[36] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, cvpr 2020. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020.

[37] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

[39] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[40] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4574–4584, 2022.

[41] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, page 371–378, New York, NY, USA, 2001. Association for Computing Machinery.