One-Shot Feed-Forward 360° Animatable Avatar via Inpainted UV-Space Gaussian Modeling

Shuling Zhao and Dan Xu✉

The Hong Kong University of Science and Technology
Zeekr Automobile R&D Co., Ltd.
szhaoax@connect.ust.hk, danxu@cse.ust.hk

Abstract. Building one-shot 3D animatable head avatars is an important yet challenging problem. Existing methods generally collapse under large camera pose variations, compromising the realism of 3D avatars. In this work, we propose a new framework to tackle the novel setting of one-shot 3D full-head animatable avatar reconstruction in a single forward pass via inpainted UV-space Gaussian modeling, enabling 360° rendering views and real-time animation. To facilitate efficient animation control, we model 3D head avatars with Gaussian primitives embedded on the surface of a parametric face model within the UV space, and project the input image features to the UV space, resulting in incomplete local UV feature maps. To inpaint the missing regions, we obtain knowledge of full-head geometry and textures from rich 3D full-head priors within a pretrained 3D generative adversarial network (GAN) for global full-head feature extraction and multi-view supervision. Specifically, to enhance the fidelity of 3D reconstruction during inpainting, we take advantage of the symmetric nature of the UV space and human faces to fuse incomplete yet detailed local UV feature maps with the extracted global full-head textures, resulting in inpainted UV Gaussian attribute maps for avatar modeling. Extensive experiments demonstrate that our method is the first to achieve high-quality 3D full-head animatable avatar modeling, significantly improving side and back views while outperforming state-of-the-art animation approaches, thereby improving the realism of 3D animatable avatars. The project page is available at https://shaelynz.github.io/fhavatar/.

Keywords: 3D Gaussian Splatting · Animatable Head Avatar · One-shot 3D Reconstruction

1 Introduction

3D animatable head avatar creation aims to reconstruct 3D animatable heads from 2D inputs, with widespread applications such as teleconferencing, virtual reality (VR), and augmented reality (AR). Despite the success achieved in 3D head reconstruction from multi-view or monocular videos [34, 48, 62, 65], the requirement for video-specific optimization makes such methods time-consuming. To improve efficiency, some approaches reconstruct 3D heads from a single image in

Fig. 1: Given a single input image, our method reconstructs animatable 3D full-head Gaussian avatars in a single forward pass. These avatars provide consistent, high-fidelity 360° view synthesis and support real-time (246 FPS) animation.

a feed-forward manner [8,9,28,43,51] using Neural Radiance Fields (NeRF) [32]. Recently, 3D Gaussian Splatting (3DGS) [19] is adopted [6,12,14] for its impressive rendering quality and speed.

Despite the advancement made in generalizable one-shot 3D animatable head avatar reconstruction, existing approaches generally fail when the rendering views significantly differ from the input camera pose. Trained on large-scale monocular face video datasets [50, 64], which predominantly capture frontal face motions, these methods are fundamentally limited to near-frontal or partial views, and cannot handle the severe ambiguity of unseen regions (*e.g.*, side/back head), which harms the realism of the reconstructed 3D heads.

To realize efficient avatar animation and 360° view synthesis at the same time, we introduce a new framework for the novel setting of one-shot feed-forward 3D full-head animatable avatar creation via inpainted UV-space Gaussian modeling. To allow for animation control, we generate Gaussian primitives in the UV space of a parametric face model (FLAME [26]) so that Gaussians are rigged to the mesh surface and can move along with it during animation. Since the input image only provides appearance information from a single view, projecting its features to the UV space results in incomplete local UV feature maps. To obtain knowledge of full-head geometry and textures, inspired by the recent success of 3D GANs in full-head generation [1,24], we utilize a pretrained 3D GAN [1] and its feed-forward inversion method [2] to efficiently extract UV-space global full-head features from the input image and provide multi-view supervision during training. As 3D GAN inversion has limited reconstruction fidelity, we inpaint the incomplete but fine-grained local UV features with the structured global UV full-head feature. Specifically, to make full use of the input view, we leverage the symmetric nature of human faces and the UV space with a transformer-based feature fusion architecture. This approach allows tokens from the global UV map to query symmetric positions on the incomplete local UV feature maps for input appearances, guiding the recovery of local details, which results in inpainted UV features for Gaussian avatar modeling. To further improve the 3D visual quality of the reconstructed avatar, we propose a 3D total variation loss that encourages Gaussians to fully cover the whole avatar surface, alleviating hole-like artifacts.

Extensive experiments confirm that our method not only is the first to reconstruct 3D full-head avatars that can be viewed in 360°, but also surpasses previous methods in avatar animation performance. Our contributions are summarized as follows:

- We propose a new framework for the novel setting of one-shot feed-forward 360° animatable avatar creation, which inpaints incomplete input local appearance details using global full-head priors from a pretrained 3D GAN in the UV space of a parametric face model for full-head Gaussian modeling.
- We perform symmetric inpainting on the local UV feature maps by taking advantage of the symmetric nature of human faces and the UV space, effectively enhancing the input appearance for high-fidelity 3D reconstruction.
- Extensive experiments on large-scale video datasets [50, 58] confirm that our method is the first to produce one-shot feed-forward 360° animatable avatars, with clear improvements in side and back view synthesis while achieving state-of-the-art performance in avatar animation.

2 Related Work

2D Talking Head Generation. By leveraging powerful 2D generative models [11, 15], promising advancements have been made in 2D talking head generation by adding motion signals into them. Some works estimate motion in an unsupervised manner, representing it with warping fields described by unsupervised keypoints [16, 39, 41, 59, 60] or latent features [44, 47]. Others adopt pretrained models to predict facial landmarks [13,30,55]. However, the lack of 3D face modeling hinders their performance, leading to face distortion and identity leakage. Some methods [10, 35, 46, 53] extract 3D Morphable Model (3DMM) [3] parameters to introduce 3D information, but they cannot support arbitrary viewpoint rendering. In contrast, we directly model human heads in 3D to enable free-viewpoint rendering.

Optimization-based 3D Animatable Head Avatar Reconstruction. To achieve free-viewpoint rendering, a line of research reconstructs 3D animatable heads from multi-view or monocular videos of a specific person. By optimizing on videos of the same identity, these methods [25, 34, 48, 56, 61, 62, 65] generally achieve high visual quality. However, they require videos with thousands of frames for optimization, which hinders practical use. To reduce the need for lengthy video data, another line of research leverages strong priors from 3D GANs [1, 5, 24, 40] and diffusion models [38]. Some methods [42, 52, 63] generate multi-view data with different motions from a single image and optimize on the generated data. Others perform optimization-based GAN inversion [31, 37, 66] from the input image on pretrained 3D GANs to create animatable 3D heads. However, the required optimization process remains time-consuming. In contrast to these methods, we create one-shot 3D animatable heads in a feed-forward manner, avoiding the cumbersome optimization process.

Feed-Forward 3D Animatable Head Avatar Reconstruction. To improve the generalizability, one-shot feed-forward 3D head reconstruction has gained

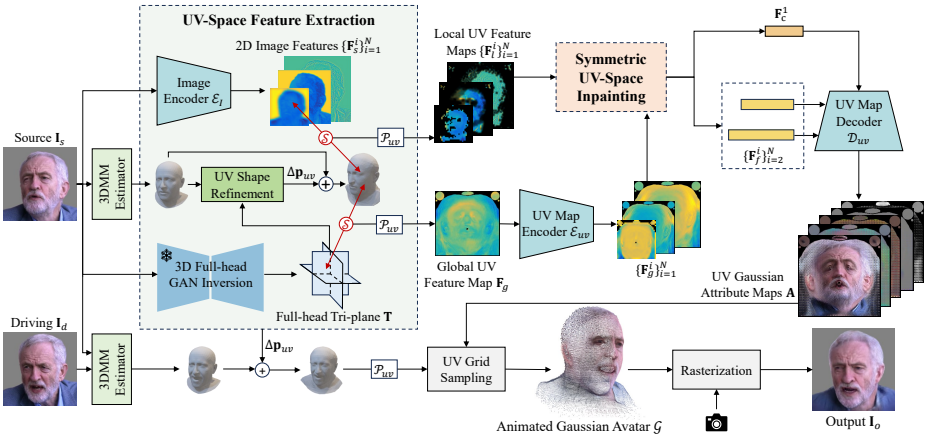

Fig. 2: Overview of the framework. Given an input source image, the UV-space feature extraction module extracts its global and local UV feature maps for animatable 3D full-head reconstruction. The symmetric UV-space inpainting module takes advantage of the symmetry of human faces and the UV space to inpaint the incomplete local UV feature maps with global features. From the predicted UV Gaussian attribute maps, 3D Gaussian primitives are sampled, which can be animated with a parametric face model and rendered given a camera pose.

increasing attention. ROME [20] extends 3DMM [26] meshes to model non-facial parts. With the excellent rendering quality of NeRF [32], some methods [8, 9, 27, 28, 43, 51] incorporate motion features into efficient tri-planes [5] for high-fidelity head avatar creation. However, their rendering speed generally remains slow. In recent years, 3DGS [19] is widely leveraged for its extraordinary rendering quality and efficiency. GAGAvatar [6] generates two sets of Gaussians for appearance and expression, and adds a neural renderer to enhance details. LAM [14] initializes Gaussians on FLAME [26] vertices and generates Gaussian attributes by querying image features through a transformer. However, trained on large monocular video datasets [50,64], which rarely contain frames of the side and back of the head, these methods generally fail when the rendering viewpoint significantly differs from the input view. Avat3r [23] and FastGHA [18] leverage multi-view inputs and Vision Transformers to infer multi-view correspondence for high-quality reconstruction. On the contrary, we aim for 360° renderings from a single image. Recent FlexAvatar [21] unifies training across monocular and multi-view datasets with learnable data source tokens, enabling the creation of complete 3D heads. In contrast, we directly inpaint the incomplete input local UV-space features with full-head priors from a pretrained 3D GAN [1], thus generating complete UV Gaussian attribute maps for full-head modeling.

3 Method

3.1 Overview

Given an input source image of a human face, denoted as $\mathbf{I}_s$, we aim to reconstruct an animatable 3D full-head Gaussian avatar on the FLAME [26] mesh

surface via UV parameterization. Fig. 2 depicts the overview of our framework. First, the UV-space feature extraction module extracts the texture features from $\mathbf{I}_s$ and projects them into the UV space, producing a global UV feature map $\mathbf{F}_g$ and multi-scale local UV feature maps $\{\mathbf{F}_l^i\}_{i=1}^N$. It also predicts a shape offset $\Delta\mathbf{p}_{uv}$ to refine the estimated FLAME mesh shape based on $\mathbf{I}_s$. Next, the symmetric UV-space inpainting module takes advantage of the symmetric nature of human faces and the UV space, inpainting the incomplete but fine-grained local UV feature maps with global features to generate a set of UV Gaussian attribute maps, from which 3D Gaussian primitives are sampled to reconstruct the Gaussian avatar. Bound to the surface of the source FLAME mesh in canonical space, the Gaussian avatar can be animated with a novel FLAME expression derived from a driving image $\mathbf{I}_d$ and rendered given a camera pose to obtain the output image $\mathbf{I}_o$. Details on UV-space Gaussian modeling, the design of the two modules, the generation of UV Gaussian attribute maps, the regularization and training strategy are described in the following subsections.

3.2 UV-Space Gaussian Modeling

To utilize the animatable FLAME model for 3D full-head reconstruction and animation, we follow [22, 54] and bind 3D Gaussian primitives to the FLAME mesh surface via UV parameterization, enabling the use of efficient 2D backbones and regularizing Gaussian positions. Specifically, we generate a UV Gaussian attribute map $\mathbf{A}_\star \in \mathbb{R}^{K\times K\times d_\star}$ for each Gaussian attribute $\star \in \{\text{color}, \text{rotation}, \text{scale}, \text{opacity}, \text{position}\}$, where K is the size of the UV map and $d_\star$ is the Gaussian attribute dimension, forming a set of UV Gaussian attribute maps $\mathbf{A} \in \mathbb{R}^{K\times K\times 14}$. Since each valid texel in the UV space already corresponds to a 3D position, we use $\mathbf{p} \in \mathbb{R}^{K\times K\times 3}$ to represent the FLAME-based UV-space 3D position map, and $\mathbf{A}_{\text{position}}$ to represent the 3D position offset. To balance the size differences between mesh faces in the UV space and in 3D space, we rescale the UV Gaussian scale map. Specifically, we compute the relative scales between the mesh face sizes on the 3D mesh surface and their corresponding UV face sizes as a relative scaling map $\mathbf{s} \in \mathbb{R}^{K\times K\times 1}$. The UV Gaussian scale map is further updated as $\mathbf{A}_{\text{scale}} = \mathbf{s} \odot \mathbf{A}_{\text{scale}}$, where $\odot$ is the Hadamard product.

To produce 3D Gaussian primitives $\mathcal{G}$, we perform grid sampling on $\mathbf{A}$ following [22, 54]. The process is formulated as:

$$\mathcal{G} = \text{grid\_sample}(\mathbf{A}, \mathcal{X}), \tag{1}$$

where $\mathcal{X}$ is the set of sampling positions in the UV space.

3.3 UV-Space Feature Extraction

To generate UV Gaussian attribute maps, we first extract the UV-space feature maps from the source image $\mathbf{I}_s$. The extracted UV feature maps are utilized for later integration.

Full-Head Feature Generation. Since $\mathbf{I}_s$ only provides the head appearance from the source camera view, it is insufficient to reconstruct the full-head appearance using only information from $\mathbf{I}_s$. To obtain knowledge from the invisible

areas in $\mathbf{I}_s$, we propose to utilize the full-head prior of a 3D full-head GAN named PanoHead [1] and its feed-forward inversion method [2] to map $\mathbf{I}_s$ to a full-head tri-plane $\mathbf{T}$, which provides coarse 3D full-head features for $\mathbf{I}_s$.

UV Shape Refinement. Utilizing the source FLAME mesh predicted by the 3DMM estimator, we can directly sample the tri-plane feature for each 3D position corresponding to a UV-space texel, producing a UV tri-plane feature map $\mathbf{F}_{\mathbf{T}}^{\mathbf{p}}$. However, FLAME meshes often fail to model the shape of the hair, making the 3D position-based feature sampling inaccurate. We propose to use a 2D UNet $\mathcal{F}_{refine}$ to refine the mesh shape. Given $\mathbf{F}_{\mathbf{T}}^{\mathbf{p}}$ and the UV-space 3D position map $\mathbf{p}$, we predict the UV shape offset $\Delta\mathbf{p}_{uv}$ as:

$$\Delta\mathbf{p}_{uv} = \mathcal{F}_{refine}([\mathbf{F}_{\mathbf{T}}^{\mathbf{p}}, \mathbf{p}]), \tag{2}$$

where $[\cdot,\cdot]$ is the concatenation operation. Then, by adding $\Delta\mathbf{p}_{uv}$ to the original 3D position map $\mathbf{p}$, we obtain a refined 3D position map $\mathbf{p}_r = \mathbf{p} + \Delta\mathbf{p}_{uv}$.

Global UV Feature Extraction. With the refined 3D position map $\mathbf{p}_r$, we directly sample the full-head tri-plane $\mathbf{T}$ to produce the global UV feature map $\mathbf{F}_g$. Produced by the pretrained 3D GAN with rich full-head prior knowledge, the tri-plane feature $\mathbf{T}$ contains convincing global head geometry and full-head textures for $\mathbf{I}_s$. Therefore, $\mathbf{F}_g$ contains a complete full-head representation of $\mathbf{I}_s$. However, constrained by the latent code dimension of the GAN inversion method and the resolution of the tri-plane, $\mathbf{T}$ cannot faithfully preserve the appearance details visible in $\mathbf{I}_s$, and consequently, neither can $\mathbf{F}_g$. Additional local features from $\mathbf{I}_s$ are needed for high-fidelity avatar reconstruction.

Local UV Feature Extraction. To extract appearance details from $\mathbf{I}_s$, we use a CNN-based encoder to encode $\mathbf{I}_s$ into multi-scale 2D image features $\{\mathbf{F}_s^i\}_{i=1}^N$. To map the image features to the UV space, we utilize the refined 3D position map $\mathbf{p}_r$. By projecting the 3D positions in $\mathbf{p}_r$ onto the 2D feature space according to the source camera pose, we can sample the corresponding 2D image feature for each UV texel, resulting in a set of multi-scale UV source features $\{\mathbf{F}_{s,uv}^i\}_{i=1}^N$. However, due to occlusion, only part of the 3D positions in $\mathbf{p}_r$ are visible in the source image. To avoid obtaining mismatched features in the UV space, we filter out invisible 3D positions in $\mathbf{p}_r$ following [17,49], which results in a UV visibility mask $\mathbf{M}_v$. By applying this mask to $\{\mathbf{F}_{s,uv}^i\}_{i=1}^N$, we exclude textures sampled from the invisible positions in the source image. The resulting multi-scale local UV feature maps $\{\mathbf{F}_l^i\}_{i=1}^N$ preserve the local appearance details from $\mathbf{I}_s$, but they are incomplete for full-head modeling.

3.4 Symmetric UV-Space Inpainting

In this module, we aim to inpaint the fine-grained local UV feature maps $\{\mathbf{F}_l^i\}_{i=1}^N$ with the coarse but complete global UV feature map $\mathbf{F}_g$ for UV Gaussian attribute map generation.

To better inpaint the multi-scale local UV feature maps $\{\mathbf{F}_l^i\}_{i=1}^N$ with the global UV feature map $\mathbf{F}_g$, we also encode $\mathbf{F}_g$ with a CNN-based UV map encoder to produce multi-scale global UV feature maps $\{\mathbf{F}_g^i\}_{i=1}^N$, and inpaint

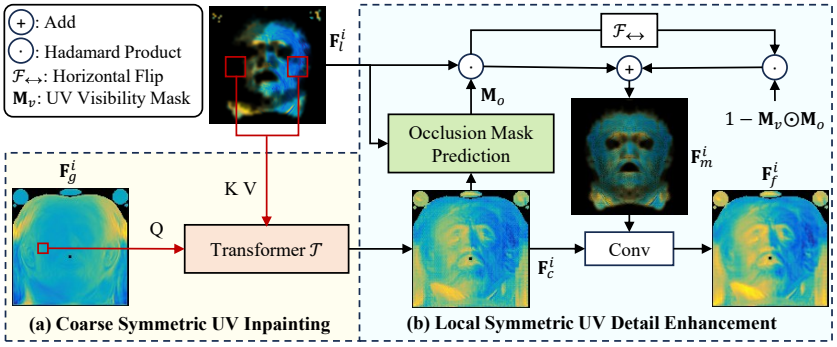

Fig. 3: Illustration of symmetric UV-space inpainting at scale i. (a) Coarse Symmetric UV Inpainting: For each feature patch in $\mathbf{F}_g^i$, we query two symmetric local windows corresponding to the patch position in $\mathbf{F}_l^i$. (b) Local Symmetric UV Detail Enhancement: The output feature $\mathbf{F}_c^i$ from (a) is further enhanced by the local UV feature map and its symmetry with convolution.

$\mathbf{F}_l^i$ with $\mathbf{F}_g^i$ at scale i across N scales. Fig. 3 illustrates symmetric UV-space inpainting at scale i.

Coarse Symmetric UV Inpainting. As a global full-head feature, $\mathbf{F}_g^i$ provides plausible texture for each valid texel in the UV space, even for the invisible area in the source view. However, it lacks the appearance details visible in the source image, which are contained in $\mathbf{F}_l^i$. Therefore, to inpaint $\mathbf{F}_l^i$, we can transfer appearance details from $\mathbf{F}_l^i$ to $\mathbf{F}_g^i$, which will result in a completed UV-space feature. Intuitively, $\mathbf{F}_l^i$ and $\mathbf{F}_g^i$ are inherently aligned, as each valid UV texel in $\mathbf{F}_l^i$ and $\mathbf{F}_g^i$ corresponds to the same 3D position on human heads. However, we generate the global and local UV feature maps using 3D position-based sampling and 3D projection, and the 3D positions and projections may still contain errors. To alleviate the potential misalignment issue caused by the errors, inspired by [33], we use a transformer architecture to fuse the features, where $\mathbf{F}_g^i$ serves as the query while $\mathbf{F}_l^i$ serves as the key and value at the cross-attention layer. As illustrated in Fig. 3 (a), for each patch token in $\mathbf{F}_g^i$, we query tokens located in a local window of size $w \times w$ centered at the corresponding patch token in $\mathbf{F}_l^i$, allowing for small deviations in 3D positions and projections. However, if the source image contains a side view, tokens on the opposite side of the face in the UV space can hardly retrieve useful information from $\mathbf{F}_l^i$, as its corresponding local window only contains the masked-out area. Essentially, human faces are generally symmetric, and the face UV space is also symmetric. To better utilize the symmetry and retrieve more useful information from $\mathbf{F}_l^i$, we also use patch tokens in the symmetric local window of $\mathbf{F}_l^i$ as key-value pairs for cross-attention.

Local Symmetric UV Detail Enhancement. As patchification inevitably leads to information loss, especially for large patches used in large-resolution feature maps at large scale i, appearance details contained in $\{\mathbf{F}_l^i\}_{i=2}^N$ may not be fully transferred to the transformer outputs $\{\mathbf{F}_c^i\}_{i=2}^N$. To further enhance the local appearance details from $\{\mathbf{F}_l^i\}_{i=2}^N$ within the aligned areas, we predict an

occlusion mask $\mathbf{M}_o$ based on $\mathbf{F}_c^i$ and $\mathbf{F}_l^i$ to mask out the region of erroneous 3D projections in $\mathbf{F}_l^i$, producing a more accurate local feature $\mathbf{F}_{l,m}^i$:

$$\mathbf{F}_{l,m}^i = \mathbf{M}_o \odot \mathbf{F}_l^i, \tag{3}$$

where $\odot$ is the Hadamard product. To utilize the symmetric nature of faces and the UV space, we horizontally flip the masked $\mathbf{F}_{l,m}^i$ and add it to the masked-out area in $\mathbf{F}_{l,m}^i$, taking full advantage of the appearance information from $\mathbf{F}_l^i$. The output $\mathbf{F}_m^i$ is further fused with $\mathbf{F}_c^i$ with a convolution layer to produce the final $\mathbf{F}_f^i$. The process is formulated as:

$$\mathbf{F}_m^i = \mathbf{F}_{l,m}^i + \mathcal{F}_{\leftrightarrow}(\mathbf{F}_{l,m}^i) \odot (1 - \mathbf{M}_v \odot \mathbf{M}_o), \tag{4}$$

$$\mathbf{F}_f^i = \mathrm{Conv}([\mathbf{F}_c^i, \mathbf{F}_m^i]), \tag{5}$$

where $\mathcal{F}_{\leftrightarrow}$ is the horizontal flip operation, $\odot$ is the Hadamard product, and $[\cdot, \cdot]$ is the concatenation operation.

3.5 UV Gaussian Attribute Map Generation

We generate the set of UV Gaussian attribute maps $\mathbf{A}$ with the UV map decoder $\mathcal{D}_{uv}$ and the inpainted UV features $\mathbf{F}_c^1$ and $\{\mathbf{F}_f^i\}_{i=2}^N$. We use the transformer output $\mathbf{F}_c^1$ instead of $\mathbf{F}_f^1$, as $\mathbf{F}_l^1$ has the smallest resolution and contains few appearance details. $\mathbf{F}_c^1$ serves as the initial input to $\mathcal{D}_{uv}$, where it is gradually upsampled and processed with residual blocks. $\{\mathbf{F}_f^i\}_{i=2}^N$ is added to the intermediate feature maps with the same size in $\mathcal{D}_{uv}$. The final output of $\mathcal{D}_{uv}$ is $\mathbf{A}$. We also generate another attribute map $\mathbf{A}^1$ only using $\mathbf{F}_c^1$ as the input, which is also used to sample Gaussian attributes $\mathcal{G}^1$ and render an output image $\mathbf{I}_o^1$ during training, preventing $\mathcal{D}_{uv}$ from relying too much on high-resolution inputs.

3.6 Regularization and Training Strategy

3D Total Variation Loss. As shown in Fig. 4, we observe that Gaussian primitives modeled on the FLAME mesh surface normally cannot cover the whole avatar surface, resulting in holes that make the Gaussians representing the opposite side of the head visible, which harms 3D reconstruction quality. To alleviate such artifacts, inspired by the UV total variation (TV) loss in [22] that uses TV loss on the UV renderings $\mathbf{I}_{uv}$ rendered from the same Gaussians $\mathcal{G}$ but with $\mathcal{G}_{\text{color}}$ set to the UV coordinates, we propose to use the TV loss on the 3D renderings $\mathbf{I}_{3d}$, which is more suitable for the UV-space definitions where neighboring pixels in the rendered image can be far from each other in the UV space (*e.g.*, eyeballs and eyelids). Specifically, based on the sampled Gaussians $\mathcal{G}$, we change the value of $\mathcal{G}_{\text{color}}$ to $\mathcal{G}_{\text{position}}$ while leaving other Gaussian attributes unchanged, and perform rendering to obtain the 3D rendering $\mathbf{I}_{3d}$. The 3D total variation loss is defined as:

$$\mathcal{L}_{3d} = \mathrm{TV}\left(\frac{\mathbf{I}_{3d} - (1 - \mathbf{I}_\alpha)}{\mathbf{I}_\alpha}\right), \tag{6}$$

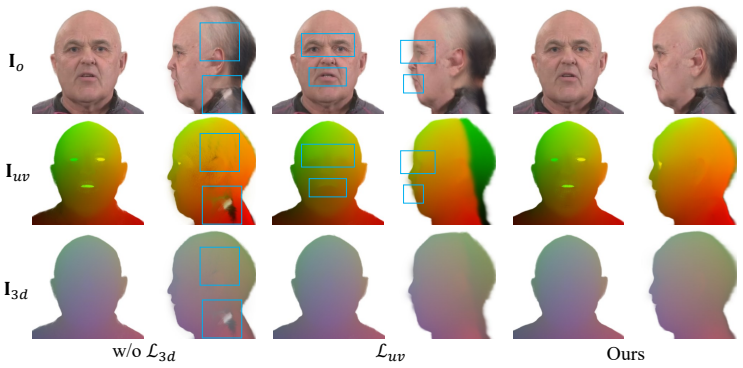

Fig. 4: Effect of the 3D total variation loss $\mathcal{L}_{3d}$. Compared with the UV total variation loss $\mathcal{L}_{uv}$ from [22], $\mathcal{L}_{3d}$ can alleviate holes on the avatar surfaces without bringing additional artifacts. Blue boxes indicate the erroneous areas in the rendered images.

where $\mathbf{I}_\alpha$ is the rendered alpha map. This regularization encourages neighboring pixels in the rendered image to be rendered from neighboring Gaussians in the 3D space.

Training Strategy. Except the offline 3DMM estimator and the frozen pretrained 3D full-head GAN and its inversion, we train the whole framework end-to-end. During training, we randomly pick two frames from a video of the same identity, in which one frame serves as the source image and the other serves as the driving image. With the source image $\mathbf{I}_s$ and the 3DMM motion parameter from the driving image $\mathbf{I}_d$, we aim to reconstruct a target image, which is $\mathbf{I}_d$. As large-scale face video datasets generally contain frontal faces, we rely on the pretrained 3D full-head GAN and its inversion to generate pseudo multi-view images of $\mathbf{I}_d$ for supervision. However, we find that the inversion method tends to generate inconsistent invisible parts of the head given different frames of a video, providing inconsistent multi-view supervision for different $\mathbf{I}_d$ given the same $\mathbf{I}_s$, which may hamper the 3D reconstruction quality. Therefore, we follow [8] to randomly switch between the animation mode and the 3D reconstruction mode. Specifically, in animation mode, $\mathbf{I}_s$ serves as the source image and $\mathbf{I}_d$ as the target image. In 3D reconstruction mode, we assign $\mathbf{I}_d$ as the source image, $\mathbf{I}_d$ together with its inverted novel views as the target images to avoid the inconsistency in the pseudo supervision.

For the training objective, we render RGB images $\mathbf{I}_o$ and $\mathbf{I}_o^1$, alpha maps $\mathbf{I}_\alpha$ and $\mathbf{I}_\alpha^1$ from $\mathcal{G}$ and $\mathcal{G}^1$, respectively. We supervise the RGB images using the target image(s) with a reconstruction loss $\mathcal{L}_{re} = \mathcal{L}_1 + \mathcal{L}_{lpips}$, and supervise the alpha maps with $\mathcal{L}_1$ loss, denoted as $\mathcal{L}_\alpha$. When $\mathbf{I}_o$ contains frontal faces, we also use an ID similarity loss [7] $\mathcal{L}_{id}$.

Apart from the 3D total variation loss, we also apply a series of regularizations to constrain the Gaussian primitives for better rendering quality. Following [54], we apply the eyeball TV loss $\mathcal{L}_{eye}$ and the position offset regularization $\mathcal{L}_{pos} = ||\mathbf{A}_{\text{position}}||_2$. To limit the extent of FLAME mesh offset, we regularize

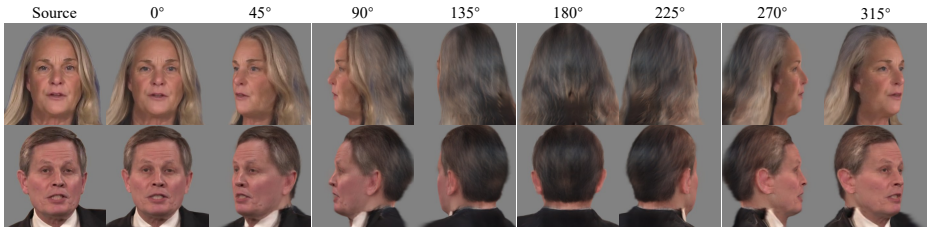

Fig. 5: Multi-view results of our method on the HDTF [58] dataset. Our avatars can be viewed in 360°.

Table 1: Quantitative comparison of our method against state-of-the-art approaches on the VFHQ dataset [50].

Method	Self-reenactment							Cross-identity Reenactment		
	PSNR ↑	SSIM ↑	LPIPS ↓	CSIM ↑	AKD ↓	AED ↓	APD ↓	CSIM ↑	AED ↓	APD ↓
StyleHEAT [53]	18.96	0.7505	0.3821	0.1695	20.78	0.7236	4.1380	0.1096	0.8195	3.3018
GOHA [28]	19.87	0.7322	0.2769	0.6469	3.746	0.4279	1.0326	0.4712	0.6859	1.9107
Real3DPortrait [51]	21.00	0.7572	0.2915	0.7696	3.934	0.3918	1.1890	0.6562	0.7186	1.7875
Portrait4D [8]	19.33	0.7166	0.3799	0.7339	6.239	0.4116	1.7931	0.5812	0.7147	2.2928
Portrait4D-v2 [9]	20.66	0.7421	0.2719	0.8075	5.367	0.3539	1.6181	0.6728	0.6625	2.2002
GAGAvatar [6]	21.60	0.7745	**0.2249**	**0.8459**	2.954	**0.3125**	0.8898	0.6681	**0.6409**	**1.7486**
LAM [14]	21.67	0.7756	0.2716	0.6846	3.676	0.4535	1.4286	0.5562	0.7271	3.2028
Ours	**23.24**	**0.7995**	0.2384	0.8012	**2.798**	0.3634	**0.8660**	**0.6757**	0.7103	2.3661

the predicted UV shape offset $\Delta\mathbf{p}_{uv}$:

$$\mathcal{L}_{shape} = \max(||\Delta\mathbf{p}_{uv}||_2 - \epsilon, 0), \tag{7}$$

where ϵ is a threshold. To avoid producing Gaussian outliers, we further add a TV regularization to $\Delta\mathbf{p}_{uv}$:

$$\mathcal{L}^{tv}_{shape} = \mathrm{TV}(\Delta\mathbf{p}_{uv}). \tag{8}$$

Thus, our regularization is defined as:

$$\mathcal{L}_{reg} = \lambda_{3d}\mathcal{L}_{3d} + \lambda_{eye}\mathcal{L}_{eye} + \lambda_{pos}\mathcal{L}_{pos} + \lambda_{shape}\mathcal{L}_{shape} + \lambda^{tv}_{shape}\mathcal{L}^{tv}_{shape}, \tag{9}$$

where λ_* is the regularization weight.

To summarize, our training objective is:

$$\mathcal{L} = \mathcal{L}_{re} + \mathcal{L}_{\alpha} + \lambda_{id}\mathcal{L}_{id} + \lambda^1(\mathcal{L}^1_{lpips} + \mathcal{L}^1_{\alpha}) + \mathcal{L}_{reg}, \tag{10}$$

where λ_{id} and λ^1 are the loss weights, and $\mathcal{L}^1_{lpips}$ and $\mathcal{L}^1_{\alpha}$ denote the losses for $\mathbf{I}^1_o$ and $\mathbf{I}^1_{\alpha}$, respectively.

4 Experiments

4.1 Experimental Setup

Datasets. We train our model on the VFHQ dataset [50]. For data preprocessing, we follow [1,2] to estimate the camera pose, crop and remove the background

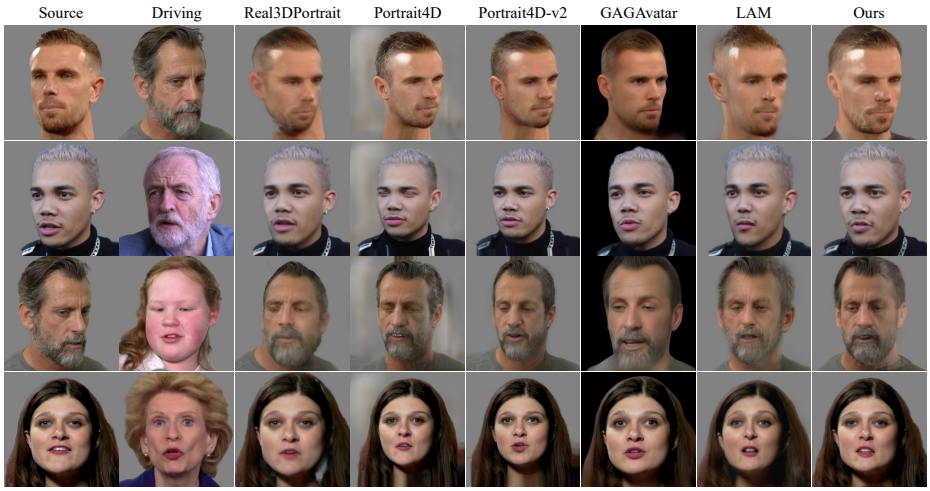

Fig. 6: Qualitative comparison with state-of-the-art methods (*i.e.*, Real3DPortrait [51], Portrait4D [8], Portrait4D-v2 [9], GAGAvatar [6] and LAM [14]) for cross-identity reenactment on the VFHQ [50] and HDTF [58] datasets. Our method best maintains source identity while effectively mimicking the driving motion.

Table 2: Quantitative comparison of our method against state-of-the-art approaches on the HDTF dataset [58].

Method	Self-reenactment							Cross-identity Reenactment		
	PSNR ↑	SSIM ↑	LPIPS ↓	CSIM ↑	AKD ↓	AED ↓	APD ↓	CSIM ↑	AED ↓	APD ↓
StyleHEAT [53]	21.91	0.8102	0.2979	0.5096	6.564	0.4860	1.1539	0.4624	0.7656	1.4266
GOHA [28]	21.33	0.7739	0.2436	0.7541	3.164	0.4128	0.7296	0.7471	**0.7287**	1.2370
Real3DPortrait [51]	23.39	0.8110	0.2362	0.8570	3.104	0.3413	0.7633	0.8886	0.7487	**1.1770**
Portrait4D [8]	21.36	0.7801	0.3245	0.8053	4.444	0.3785	1.1675	0.7876	0.7483	1.4278
Portrait4D-v2 [9]	22.72	0.7985	0.2197	0.8641	4.278	0.3392	1.1729	0.8668	0.7333	1.4197
GAGAvatar [6]	23.72	0.8177	0.2089	**0.8894**	2.679	**0.3034**	0.5977	**0.9004**	0.7300	1.2045
LAM [14]	23.55	0.8167	0.2362	0.7542	3.386	0.4180	0.8911	0.7656	0.7731	1.6635
Ours	**26.61**	**0.8642**	**0.1900**	0.8622	**2.287**	0.3318	**0.5286**	0.8568	0.7347	1.5605

of each video frame. All frames are resized to 512×512. To obtain the FLAME parameters, we utilize the head tracker from [34]. For performance evaluation, we use the VFHQ test split. To evaluate the generalizability, we randomly sample 20 videos from HDTF dataset [58] following [6,14]. We also sample 21 videos from a multi-view MEAD [45] video dataset to assess the effectiveness of our design under large camera pose changes.

Metrics. For self-reenactment, given ground-truth frames, we employ PSNR, SSIM, LPIPS [57] to measure the reconstruction quality. For identity preservation, we use cosine similarity (CSIM) of identity features extracted by [7]. For driving accuracy, we use average keypoint distance (AKD) based on [4], average expression distance (AED) based on [36] and average pose distance (APD) based on [29]. For the setting of cross-identity reenactment, with no ground-truth available, we use CSIM, AED and APD for evaluation, following prior works [6, 14].

Table 3: Animation speed measured in FPS. We exclude the time for source avatar reconstruction and driving motion extraction that can be computed beforehand, and take an average over 100 frames.

Method	StyleHEAT [53]	GOHA [28]	Real3DPortrait [51]	Portrait4D-v2 [9]	GAGAvatar [6]	LAM [14]	Ours
FPS	2.19	4.91	11.02	18.39	58.11	231.74	**246.00**

4.2 Comparison with State-of-the-art Methods

We compare with 2D-based StyleHEAT [53], and 3D-based GOHA [28], Real3DPortrait [51], Portrait4D [8], Portrait4D-v2 [9], GAGAvatar [6] and LAM [14].

Quantitative Comparison. We report the results of the quantitative comparison with state-of-the-art methods on the VFHQ [50] and HDTF [58] datasets in Tab. 1 and Tab. 2. For self-reenactment, we utilize the initial frame from each video as the source image and the remainder as the driving images. To perform cross-identity reenactment, we randomly sample a frame from the video of a different identity as the source image. Our method generally achieves the best performance for self-reenactment and remains competitive for cross-identity reenactment. With respect to reconstruction quality (*i.e.*, PSNR, SSIM), our method significantly outperforms previous methods on both datasets, highlighting the effectiveness of symmetric UV-space inpainting, where appearance details from the input source image are fully preserved for Gaussian attribute map generation. Our method also achieves the best AKD and APD for self-reenactment, which confirms the accuracy of expression and pose during animation. For cross-identity reenactment, our method outperforms others in identity preservation on the VFHQ dataset, where large differences in camera pose between source and driving image pairs occur most frequently. This indicates the superiority of modeling 3D full heads. In terms of motion accuracy for cross-identity reeanctment, we achieve slightly worse AED and APD. Based on the motion from the tracked FLAME parameters, the expression and pose accuracy of our method is limited by the tracking accuracy. Nevertheless, for AED and APD scores, our method outperforms LAM [14], which shares the same FLAME head tracker with us.

Qualitative Comparison. Fig. 6 presents a qualitative comparison of our cross-identity reenactment results against state-of-the-art methods. Our method better maintains source face identity and head shape than previous methods under large camera pose changes (rows 1 and 3), while accurately imitating driving expressions (rows 2 and 4). This confirms the importance of 3D full-head reconstruction during animation. The 360° rendering results of the avatars reconstructed by our method in Fig. 5 validate the effectiveness of our UV-space full-head modeling, allowing for more realistic 3D avatars. More qualitative comparison results are shown in the supplementary material.

Animation Speed. Tab. 3 measures the time of reenactment during inference using an NVIDIA RTX 3090. Based on the efficient 3DGS representation, our method achieves an FPS of 246, outperforming previous methods. This confirms the efficiency of our method and highlights its potential real-time applications.

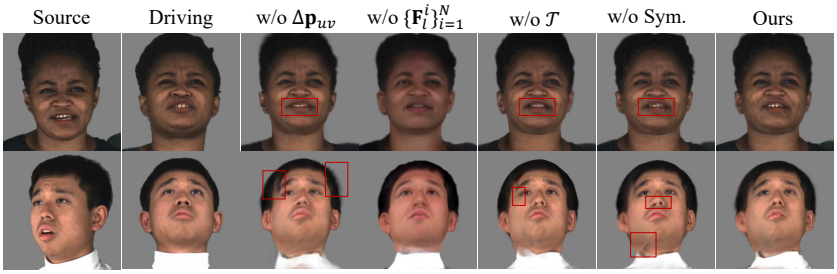

Fig. 7: Qualitative ablation study on the MEAD [45] dataset. Our method performs best with clearer facial features and less artifacts.

Table 4: Ablation study on the design of our framework. We present the results for self-reenactment on the HDTF [58] and multi-view MEAD [45] datasets. Our complete design achieves consistently the best results on both datasets.

Method	HDTF							MEAD						
	PSNR ↑	SSIM ↑	LPIPS ↓	CSIM ↑	AKD ↓	AED ↓	APD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	CSIM ↑	AKD ↓	AED ↓	APD ↓
w/o $\Delta \mathbf{p}_{uv}$	26.15	0.8596	0.2058	0.8618	2.338	0.3321	0.5751	17.09	0.7495	0.3493	0.6678	5.887	0.5032	2.6566
w/o $\{\mathbf{F}_l^i\}_{i=1}^N$	23.82	0.8186	0.2658	0.5230	2.732	0.3986	0.7115	17.09	**0.7508**	0.3541	0.3932	4.849	0.6903	1.9517
w/o $\mathcal{T}$	25.79	0.8579	0.2049	0.8461	2.335	0.3443	0.5831	17.05	0.7468	0.3430	0.6500	**4.792**	0.5044	2.1080
w/o Sym.	26.60	**0.8651**	**0.1893**	**0.8646**	2.307	0.3402	0.5587	17.18	0.7475	**0.3402**	0.6631	5.103	0.5047	2.2145
$\mathcal{L}_{uv}$	26.40	0.8622	0.1911	0.8567	2.324	0.3447	0.5370	17.08	0.7459	0.3458	0.6663	5.058	**0.4806**	1.9627
Ours	**26.61**	0.8642	0.1900	0.8622	**2.287**	**0.3318**	**0.5286**	**17.20**	0.7495	**0.3402**	**0.6765**	4.844	0.5037	**1.9484**

4.3 Ablation Study

We perform ablation studies to analyze the effectiveness of our framework design. The quantitative comparison is presented in Tab. 4 and the qualitative results are depicted in Fig. 7, Fig. 8 and Fig. 4.

UV Shape Refinement. According to Tab. 4, removing UV shape refinement (w/o $\Delta \mathbf{p}_{uv}$) degrades the overall performance, as we cannot obtain a more accurate 3D position to sample the appropriate features from the full-head tri-plane and the 2D image features. Fig. 7 shows blurry teeth and hair as well as clear boundary artifacts in the rendered frames without UV shape refinement.

Local UV Feature Maps. Tab. 4 and Fig. 7 show that we can hardly preserve the identity from the input source image without using local UV feature maps (w/o $\{\mathbf{F}_l^i\}_{i=1}^N$), with a significant drop in CSIM. Although the full-head tri-plane is inverted from the source image, it cannot preserve appearance details, as the dimension of the inverted latent code and the tri-plane resolution are limited. We cannot reconstruct faithful 3D head avatars without using local UV feature maps, which confirms the importance of inpainting based on them.

Symmetric UV-Space Inpainting. We utilize a transformer-based UV feature inpainting strategy with a symmetry prior. We analyze its effectiveness by removing the transformer (w/o $\mathcal{T}$) and removing symmetric operations (w/o Sym.) respectively.

When the transformer is removed, the symmetric local UV features are directly fused with the global UV features without filtering out potential errors in 3D projection, which decreases the overall performance as shown in Tab. 4. We can observe that the teeth are blurry and artifacts appear at the eye in Fig. 7, which is caused by inaccurate 3D projections.

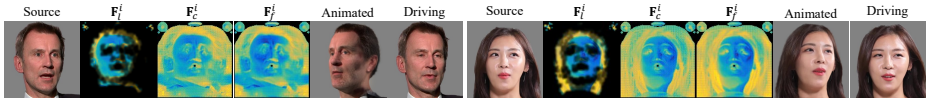

Fig. 8: Visualization of the UV-space feature maps before and after symmetric UV-space inpainting. The inpainted UV-space features support 360° animatable avatars. We present results at scale $i = 3$.

After we remove the symmetric operations, the quantitative results in Tab. 4 are generally comparable to the full design on HDTF, but drop consistently when evaluated on the multi-view MEAD dataset. Since the symmetric operations take the most effect when only a side face is visible in the source image, they cannot contribute significantly to the final results given the videos with frontal faces in HDTF. Nevertheless, they can generally improve full-head avatar animation quality on the multi-view video dataset where source and driving camera poses may vary significantly. As shown in Fig. 7 row 2, the symmetric operations reduce artifacts at the source view boundaries, improving the reconstruction quality of the occluded parts. Additionally, Tab. 4 indicates that the symmetric operations consistently improve facial motion accuracy (*i.e.*, AKD, AED, APD). The reason is that by enforcing symmetry on local UV feature maps, the accuracy of projections from 2D image features to the UV space can be improved, making the generated Gaussian attribute maps more aligned with the underlying FLAME meshes. The improved projection accuracy is also confirmed by the clearer teeth when the symmetric operations are added in Fig. 7 row 1.

Fig. 8 visualizes the UV-space feature maps before and after symmetric UV-space inpainting. Compared to the local UV map $\mathbf{F}_l^i$, the transformer output $\mathbf{F}_c^i$ provides a coarse but complete UV-space full-head representation. Further enhanced by $\mathbf{F}_l^i$, the final inpainting output $\mathbf{F}_f^i$ reduces grid-like artifacts caused by patchification in $\mathbf{F}_c^i$, preserving appearance details from the source view. The inpainted UV-space features are utilized for UV Gaussian attribute map generation, enabling 360° animatable Gaussian avatar reconstruction.

3D Total Variation Loss. As shown in Fig. 4, without using the 3D total variation loss (w/o $\mathcal{L}_{3d}$), the generated Gaussians cannot cover the avatar surface, leaving holes on the reconstructed avatars, which greatly harms the 3D quality.

Using the UV total variation loss ($\mathcal{L}_{uv}$) from [22] instead of $\mathcal{L}_{3d}$ can also encourage the Gaussians to cover the avatar surface, as $\mathcal{L}_{uv}$ encourages neighboring pixels in the rendered image to be from neighboring texels in the UV space. However, we use a different UV parametrization where eyeballs and inner mouth are far from the eyelid and lip areas in the UV space for full head modeling. Therefore, as shown in Fig. 4, using $\mathcal{L}_{uv}$ leads to artifacts in the eye and mouth areas, causing Gaussians bound to other face areas to cover the eyeballs and mouth, which harms the avatar animation quality. The degradation is also confirmed by the quantitative results in Tab. 4. In contrast, using our $\mathcal{L}_{3d}$ can realize a 3D consistent full-head avatar without damaging the animation quality.

5 Conclusion

In this paper, we have tackled a novel setting of building one-shot feed-forward 3D full-head animatable avatars, which allows for more immersive 3D talking heads. We propose a novel framework that generates complete Gaussian attribute maps in the UV space of a parametric head model via inpainting. Specifically, we perform symmetric UV-space inpainting on the incomplete but fine-grained input local image features with the rich 3D full-head prior knowledge from a pretrained 3D GAN, effectively enhancing the fidelity of 3D reconstruction. Extensive experiments confirm that our method is able to generate one-shot feed-forward 3D animatable head avatars that can be viewed in 360°.

Acknowledgements. The research is supported in part by Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321, ITF PRP/046/24FX, Science & Technology Cooperation Program of Shandong under grant No. SDST26EG01, SAIL Research Project, HKUST-Zeekr Coolaborative Research Fund.

References

1. An, S., Xu, H., Shi, Y., Song, G., Ogras, U.Y., Luo, L.: Panohead: Geometry-aware 3d full-head synthesis in 360deg. In: CVPR (2023)
2. Bilecen, B.B., Gökmen, A., Dundar, A.: Dual encoder gan inversion for high-fidelity 3d head reconstruction from single images. In: NeurIPS (2024)
3. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH (1999)
4. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In: ICCV (2017)
5. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: CVPR (2022)
6. Chu, X., Harada, T.: Generalizable and animatable gaussian head avatar. In: NeurIPS (2024)
7. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
8. Deng, Y., Wang, D., Ren, X., Chen, X., Wang, B.: Portrait4d: Learning one-shot 4d head avatar synthesis using synthetic data. In: CVPR (2024)
9. Deng, Y., Wang, D., Wang, B.: Portrait4d-v2: Pseudo multi-view data creates better 4d head synthesizer. In: ECCV (2024)
10. Doukas, M.C., Zafeiriou, S., Sharmanska, V.: Headgan: One-shot neural head synthesis and editing. In: ICCV (2021)
11. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
12. Guo, C., Su, Z., Wang, J., Li, S., Chang, X., Li, Z., Zhao, Y., Wang, G., Huang, R.: Sega: Drivable 3d gaussian head avatar from a single image. arXiv preprint arXiv:2504.14373 (2025)
13. Ha, S., Kersner, M., Kim, B., Seo, S., Kim, D.: Marionette: Few-shot face reenactment preserving identity of unseen targets. In: AAAI (2020)

14. He, Y., Gu, X., Ye, X., Xu, C., Zhao, Z., Dong, Y., Yuan, W., Dong, Z., Bo, L.: Lam: Large avatar model for one-shot animatable gaussian head. In: SIGGRAPH (2025)
15. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
16. Hong, F.T., Zhang, L., Shen, L., Xu, D.: Depth-aware generative adversarial network for talking head video generation. In: CVPR (2022)
17. Hu, S., Hong, F., Pan, L., Mei, H., Yang, L., Liu, Z.: Sherf: Generalizable human nerf from a single image. In: ICCV (2023)
18. Ji, X., Weiss, S., Kansy, M., Naruniec, J., Cao, X., Solenthaler, B., Bradley, D.: Fastgha: Generalized few-shot 3d gaussian head avatars with real-time animation. In: ICLR (2026)
19. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG **42**(4), 139–1 (2023)
20. Khakhulin, T., Sklyarova, V., Lempitsky, V., Zakharov, E.: Realistic one-shot mesh-based head avatars. In: ECCV (2022)
21. Kirschstein, T., Giebenhain, S., Nießner, M.: Flexavatar: Learning complete 3d head avatars with partial supervision. arXiv preprint arXiv:2512.15599 (2025)
22. Kirschstein, T., Giebenhain, S., Tang, J., Georgopoulos, M., Nießner, M.: Gghead: Fast and generalizable 3d gaussian heads. In: SIGGRAPH Asia (2024)
23. Kirschstein, T., Romero, J., Sevastopolsky, A., Nießner, M., Saito, S.: Avat3r: Large animatable gaussian reconstruction model for high-fidelity 3d head avatars. In: ICCV (2025)
24. Li, H., Chen, C., Shi, T., Qiu, Y., An, S., Chen, G., Han, X.: Spherehead: stable 3d full-head synthesis with spherical tri-plane representation. In: ECCV (2024)
25. Li, L., Li, Y., Weng, Y., Zheng, Y., Zhou, K.: Rgbavatar: Reduced gaussian blendshapes for online modeling of head avatars. In: CVPR (2025)
26. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4d scans. ACM TOG **36**(6), 1–17 (2017)
27. Li, W., Zhang, L., Wang, D., Zhao, B., Wang, Z., Chen, M., Zhang, B., Wang, Z., Bo, L., Li, X.: One-shot high-fidelity talking-head synthesis with deformable neural radiance field. In: CVPR (2023)
28. Li, X., De Mello, S., Liu, S., Nagano, K., Iqbal, U., Kautz, J.: Generalizable one-shot 3d neural head avatar. In: NeurIPS (2023)
29. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J., et al.: Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172 (2019)
30. Ma, Y., Liu, H., Wang, H., Pan, H., He, Y., Yuan, J., Zeng, A., Cai, C., Shum, H.Y., Liu, W., et al.: Follow-your-emoji: Fine-controllable and expressive freestyle portrait animation. In: SIGGRAPH Asia (2024)
31. Ma, Z., Zhu, X., Qi, G.J., Lei, Z., Zhang, L.: Otavatar: One-shot talking face avatar with controllable tri-plane rendering. In: CVPR (2023)
32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
33. Pan, P., Su, Z., Lin, C., Fan, Z., Zhang, Y., Li, Z., Shen, T., Mu, Y., Liu, Y.: Humansplat: Generalizable single-image human gaussian splatting with structure priors. In: NeurIPS (2024)
34. Qian, S., Kirschstein, T., Schoneveld, L., Davoli, D., Giebenhain, S., Nießner, M.: Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In: CVPR (2024)

35. Ren, Y., Li, G., Chen, Y., Li, T.H., Liu, S.: Pirenderer: Controllable portrait image generation via semantic neural rendering. In: ICCV (2021)
36. Retsinas, G., Filntisis, P.P., Danecek, R., Abrevaya, V.F., Roussos, A., Bolkart, T., Maragos, P.: 3d facial expressions through analysis-by-neural-synthesis. In: CVPR (2024)
37. Roich, D., Mokady, R., Bermano, A.H., Cohen-Or, D.: Pivotal tuning for latent-based editing of real images. ACM TOG **42**(1), 1–13 (2022)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
39. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: NeurIPS (2019)
40. Sun, J., Wang, X., Wang, L., Li, X., Zhang, Y., Zhang, H., Liu, Y.: Next3d: Generative neural texture rasterization for 3d-aware head avatars. In: CVPR (2023)
41. Tao, J., Gu, S., Li, W., Duan, L.: Learning motion refinement for unsupervised face animation. In: NeurIPS (2024)
42. Taubner, F., Zhang, R., Tuli, M., Lindell, D.B.: Cap4d: Creating animatable 4d portrait avatars with morphable multi-view diffusion models. In: CVPR (2025)
43. Tran, P., Zakharov, E., Ho, L.N., Tran, A.T., Hu, L., Li, H.: Voodoo 3d: Volumetric portrait disentanglement for one-shot 3d head reenactment. In: CVPR (2024)
44. Wang, D., Deng, Y., Yin, Z., Shum, H.Y., Wang, B.: Progressive disentangled representation learning for fine-grained controllable talking head synthesis. In: CVPR (2023)
45. Wang, K., Wu, Q., Song, L., Yang, Z., Wu, W., Qian, C., He, R., Qiao, Y., Loy, C.C.: Mead: A large-scale audio-visual dataset for emotional talking-face generation. In: ECCV (2020)
46. Wang, Q., Zhang, L., Li, B.: Safa: Structure aware face animation. In: 3DV (2021)
47. Wang, Y., Yang, D., Bremond, F., Dantcheva, A.: Latent image animator: Learning to animate images via latent space navigation. In: ICLR (2022)
48. Xiang, J., Gao, X., Guo, Y., Zhang, J.: Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In: CVPR (2024)
49. Xie, J., Ouyang, H., Piao, J., Lei, C., Chen, Q.: High-fidelity 3d gan inversion by pseudo-multi-view optimization. In: CVPR (2023)
50. Xie, L., Wang, X., Zhang, H., Dong, C., Shan, Y.: Vfhq: A high-quality dataset and benchmark for video face super-resolution. In: CVPRW (2022)
51. Ye, Z., Zhong, T., Ren, Y., Yang, J., Li, W., Huang, J., Jiang, Z., He, J., Huang, R., Liu, J., Zhang, C., Yin, X., MA, Z., Zhao, Z.: Real3d-portrait: One-shot realistic 3d talking portrait synthesis. In: ICLR (2024)
52. Yin, F., Yao, C.H., Mantiuk, R.K., Jampani, V., et al.: Facecraft4d: Animated 3d facial avatar generation from a single image. In: ICCV (2025)
53. Yin, F., Zhang, Y., Cun, X., Cao, M., Fan, Y., Wang, X., Bai, Q., Wu, B., Wang, J., Yang, Y.: Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan. In: ECCV (2022)
54. Yu, Z., Li, T., Sun, J., Shapira, O., Park, S., Stengel, M., Chan, M., Li, X., Wang, W., Nagano, K., et al.: Gaia: Generative animatable interactive avatars with expression-conditioned gaussians. In: SIGGRAPH (2025)
55. Zakharov, E., Ivakhnenko, A., Shysheya, A., Lempitsky, V.: Fast bi-layer neural synthesis of one-shot realistic head avatars. In: ECCV (2020)
56. Zhang, J., Wu, Z., Liang, Z., Gong, Y., Hu, D., Yao, Y., Cao, X., Zhu, H.: Fate: Full-head gaussian avatar with textural editing from monocular video. In: CVPR (2025)

57. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
58. Zhang, Z., Li, L., Ding, Y., Fan, C.: Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In: CVPR (2021)
59. Zhao, J., Zhang, H.: Thin-plate spline motion model for image animation. In: CVPR (2022)
60. Zhao, S., Hong, F.T., Huang, X., Xu, D.: Synergizing motion and appearance: Multi-scale compensatory codebooks for talking head video generation. In: CVPR (2025)
61. Zheng, Y., Abrevaya, V.F., Bühler, M.C., Chen, X., Black, M.J., Hilliges, O.: Im avatar: Implicit morphable head avatars from videos. In: CVPR (2022)
62. Zheng, Y., Yifan, W., Wetzstein, G., Black, M.J., Hilliges, O.: Pointavatar: Deformable point-based head avatars from videos. In: CVPR (2023)
63. Zhou, Z., Ma, F., Fan, H., Chua, T.S.: Zero-1-to-a: Zero-shot one image to animatable head avatars using video diffusion. In: CVPR (2025)
64. Zhu, H., Wu, W., Zhu, W., Jiang, L., Tang, S., Zhang, L., Liu, Z., Loy, C.C.: Celebv-hq: A large-scale video facial attributes dataset. In: ECCV (2022)
65. Zielonka, W., Bolkart, T., Thies, J.: Instant volumetric head avatars. In: CVPR (2023)
66. Zielonka, W., Garbin, S.J., Lattas, A., Kopanas, G., Gotardo, P., Beeler, T., Thies, J., Bolkart, T.: Synthetic prior for few-shot drivable head avatar inversion. In: CVPR (2025)