MeGA: Hybrid Mesh-Gaussian Head Avatar for High-Fidelity Rendering and Head Editing

Cong Wang[1], Di Kang[2], Heyi Sun[1], Shenhan Qian[3], Zixuan Wang[4],
Linchao Bao[2], *Song-Hai Zhang[1]

[1]Tsinghua University, [2]Tencent, [3]Technical University of Munich, [4]Carnegie Mellon University

*Corresponding author: shz@tsinghua.edu.cn

Abstract

Creating high-fidelity head avatars from multi-view videos is essential for many AR/VR applications. However, current methods often struggle to achieve high-quality renderings across all head components (e.g., skin vs. hair) due to the limitations of using one single *representation for elements with varying characteristics. In this paper, we introduce a Hybrid **Me**sh-**G**aussian Head **A**vatar (MeGA) that models different head components with more suitable representations. Specifically, we employ an enhanced FLAME mesh for the facial representation and predict a UV displacement map to provide per-vertex offsets for improved personalized geometric details. To achieve photorealistic rendering, we use deferred neural rendering to obtain facial colors and decompose neural textures into three meaningful parts. For hair modeling, we first build a static canonical hair using 3D Gaussian Splatting. A rigid transformation and an MLP-based deformation field are further applied to handle complex dynamic expressions. Combined with our occlusion-aware blending, MeGA generates higher-fidelity renderings for the whole head and naturally supports diverse downstream tasks. Experiments on the NeRSemble dataset validate the effectiveness of our designs, outperforming previous state-of-the-art methods and enabling versatile editing capabilities, including hairstyle alteration and texture editing. The code is released in* https://github.com/conallwang/MeGA.

1. Introduction

Generating photorealistic rendering of animatable head avatars has been a long-standing focus in computer vision and graphics, with applications spanning AR/VR communication [14, 24, 30], gaming [39], and remote collaborations [42].

Existing methods have explored mesh-based representations [1, 13, 24, 28], NeRF-based representations [15, 27, 40, 45], 3D Gaussians-based representations [9, 32, 41, 44] and achieved remarkable progress in this field. However, the human head is a complex "object" containing components with drastically different characteristics so there may not exist one *single* representation that can model all of them well simultaneously. For instance, the human hair contains volumetric thin structures while the human face is predominantly surface-like regions and can be animated in a low dimensional space [23]. Thus, using only one representation to model different head components inevitably sacrifices the rendering quality of one part for another.

Ideally, we expect the head avatar representation can be rendered in photorealistic quality and can be easily controlled to perform vivid facial animations. For high-quality *facial* rendering and animation, Pixel Codec Avatars (PiCA) [28], which adopts neural texture representation [38], have demonstrated extraordinary rendering quality and subtle dynamic texture details while being able to be animated easily due to its mesh-based representation. However, it contains noticeable artifacts including texture-like hair rendering and mesh-like hair boundaries. In contrast, GaussianAvatars [32], which adopts rigged 3D Gaussian Splatting (3DGS) [19] representation, successfully reconstructs high-frequency volumetric human hair but shows inferior facial texture details (e.g., wrinkles) and interpenetration artifacts (e.g., Fig. 4, first row). Additionally, anti-aliasing of 3DGS remains an open problem [22, 37, 49], significantly impairing its rendering quality on human faces, particularly when zooming in/out.

Therefore, we propose to use more suitable representations for different head components (i.e., neural mesh for the face and 3DGS for the hair), resulting in a Hybrid **Me**sh-**G**aussian Head **A**vatar (MeGA). Specifically, we adopt the FLAME mesh [23] as our base mesh to model dynamic human faces. Additionally, we learn a UV displacement map conditioned on the driving signal (i.e., FLAME parameters) to account for the geometric details that cannot be represented in the FLAME space. For photorealistic rendering,

we use neural texture and deferred neural rendering techniques [28, 38]. Unlike PiCA [28], our neural texture consists of three components, including a diffuse texture map to model the base color, an *expression-dependent* texture map to model dynamic textures (e.g., wrinkles and dimples), and a *view-dependent* texture map to handle view-dependent effects. For hair modeling, we build an canonical 3DGS hair from a chosen frame, which is subsequently deformed by a rigid transformation and an MLP network to capture dynamic hair motion.

Another crucial component of MeGA for high-quality head renderings is the occlusion-aware blending for face and hair images. Specifically, we conduct occlusion test using our "near-z" GS depths rather than commonly used integrated GS depth, enabling more stable training. To minimizing blending artifacts, we propose an early-stopping strategy during the GS hair rendering to exclude the occluded Gaussians, combined with a soft-blending technique to create smoother blending boundaries (e.g., hairline).

With this decomposed representation, MeGA not only achieves state-of-the-art rendering quality for the complete head but also supports a range of downstream operations, including hairstyle alterations and texture editing.

Our contributions are summarized below:

- We are the first to propose a hybrid mesh-Gaussian full-head representation, adopting more suitable representations to model different head components (i.e., neural mesh for the face, 3DGS for the hair).
- The decomposed representation naturally supports various downstream applications, including high-quality hair alteration and texture editing.
- Experimental results on the NeRSemble dataset show that our approach produces higher-quality renderings for novel expressions and views.

2. Related Works

2.1. Animatable Head Avatars

Creating high-fidelity, animatable 3D head avatars from images or videos has always been of great interest in the computer vision and graphics community. Traditional explicit geometric modeling methods [2, 17, 18] usually rely on low-poly meshes and suffer from inaccurate details, especially around hair regions. With the rise of neural network-based approaches, Codec Avatars [24, 26, 28, 34, 40, 43] utilize coarse tracked meshes together with neural networks to model and render facial performance sequences by capturing them from multi-view videos. The captured avatars can be animated using a driving model [24] that maps control signals to the avatar latent codes; however, this approach may lack intuitive controls. Another line of work [11–13, 16, 32, 44, 46, 50–52, 54] aims to model head avatars that can be directly driven using parameters from existing parametric models (e.g., FLAME [23]). It is noteworthy that methods utilizing multi-view video inputs [32, 44] typically significantly outperform those relying on monocular inputs [11–13, 16, 46, 50–52, 54]. Our work follows the multi-view video setting like the GaussianAvatars [32].

2.2. 3D Representations for Head Avatars

Traditional 3D head avatars [2, 17, 18] typically employ a topological consistent, morphable mesh model (e.g., 3DMM) [5, 23] for facial modeling and animation. However, it is exceedingly challenging to faithfully reconstruct the intricate details of the face and complicated hair regions using standard 3DMMs. To address these challenges, implicit head avatar models integrate neural networks into the avatar modeling and rendering processes. For instance, the Neural Head Avatar [13] and IM Avatar [51] leverage neural networks to model the geometric and texture details beyond the FLAME model [23]. The Deferred Neural Rendering [38] approach achieves high-quality, photo-realistic rendering with imperfect 3D assets by substituting the graphics rendering pipeline with a neural network-based rendering process. In addition to the mesh-based representations [4, 13, 51], there are research works based on point-based representations [40, 52], volume-based representations [25, 46], the mixture of volumetric primitives [26], NeRF-based representations [6, 11, 12, 16, 50], and more recent 3D Gaussians-based representations [9, 29, 32, 41, 44, 47]. Different from previous methods, we employ a hybrid mesh-Gaussian representation to decouple the modeling of the human face and hair.

Note that GaussianAvatars [32] only uses the mesh as the underlying deformation proxy to obtain an animatable 3DGS-based head. The potential artifacts (e.g., inferior facial details and interpenetration artifacts) of 3DGS are enlarged due to large scale (e.g., jaw open) and non-rigid deformation (e.g., extreme expressions). DELTA [10] leverages mesh and NeRF to model faces and hair separately, which is conceptually similar to our approach. However, by incorporating deferred neural rendering and advancing from NeRF to 3DGS, our MeGA achieves higher-quality renderings and greatly improved efficiency, while also supporting a broader range of downstream applications.

3. Hybrid Mesh-Gaussian Head Avatar

Our goal is to create an animatable head avatar from multi-view videos that can be driven by FLAME parameters. Specifically, As illustrated in Fig. 1, given the driving signal (i.e., FLAME shape β, expression ψ, and pose ϕ parameters) and view vector $\boldsymbol{d}$, we employ three decoders to generate a UV displacement map $\hat{\boldsymbol{G}}_d$, a view texture map $\hat{\boldsymbol{T}}_v$, and a dynamic texture map $\hat{\boldsymbol{T}}_{dy}$. The UV displacement map $\hat{\boldsymbol{G}}_d$ captures geometric details beyond the FLAME. The view texture map $\hat{\boldsymbol{T}}_v$, dynamic texture map $\hat{\boldsymbol{T}}_{dy}$ and diffuse texture map $\hat{\boldsymbol{T}}_{di}$ are combined to produce facial neural textures

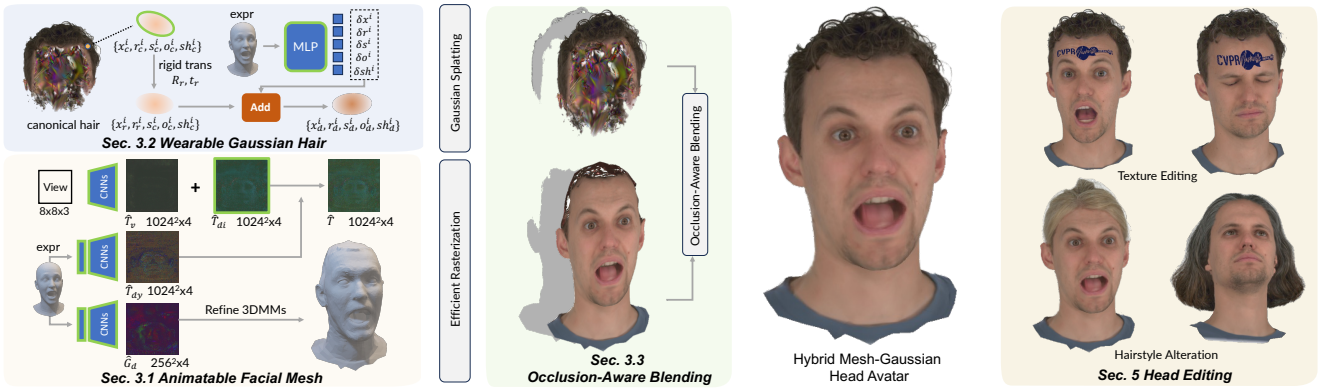

Figure 1. **Hybrid Mesh-Gaussian Head Avatar.** MeGA models different head components with more suitable representations. For *facial* modeling, we propose a neural mesh-based representation, including a UV displacement map $\hat{\boldsymbol{G}}_d$ for geometric details, a disentangled neural texture map composed by $\hat{\boldsymbol{T}}_{di}$, $\hat{\boldsymbol{T}}_{dy}$, and $\hat{\boldsymbol{T}}_v$ to learn the diffuse colors, dynamic textures, and view-dependent colors, respectively. For *hair* modeling, a canonical 3D Gaussian Splatting is reconstructed and then animated using a rigid transformation and an MLP-based non-rigid deformation field. A mesh occlusion-aware blending is proposed to properly blend the face and hair images. MeGA naturally supports hair alteration and texture editing due to the disentangled representations. Learnable parameters are highlighted using green boxes.

$\hat{\boldsymbol{T}}$. Facial colors are then obtained through efficient mesh rasterization, followed by a lightweight per-pixel decoder. For hair modeling, we create a static canonical 3DGS hair from a chosen frame and incorporate a global rigid transformation and an MLP-based non-rigid deformation field for animation. Finally, a mesh occlusion-aware blending is proposed to properly blend the face and hair images.

3.1. Animatable Facial Mesh

To precisely control head avatars and robust generalization to unseen expressions, we use an enhanced FLAME mesh as our facial geometry, along with a UV displacement map to capture personalized geometric details. Disentangled neural textures are mapped onto this refined facial mesh and decoded into RGB colors via our per-pixel texture decoder.

Enhanced FLAME Mesh. To increase the expressiveness of FLAME mesh, similar to [13], we densify the FLAME mesh using four-way subdivision and add faces for human teeth, generating our enhanced FLAME mesh:

$$\mathcal{T}(\beta, \psi, \phi) = \{\mathcal{V}(\beta, \psi, \phi), \mathcal{F}\}, \tag{1}$$

where $\mathcal{V} \in \mathbb{R}^{16428\times 3}$ represents the vertices of the enhanced mesh, calculated using the shape $\beta \in \mathbb{R}^{300}$, expression $\psi \in \mathbb{R}^{100}$, and pose $\phi \in \mathbb{R}^{15}$ parameters via linear blend skinning (LBS). The faces of the enhanced mesh are denoted by $\mathcal{F} \in \mathbb{R}^{40212\times 3}$.

Geometry Refinement. Building on the enhanced FLAME mesh, and inspired by [36, 40], we predict a UV displacement map $\hat{\boldsymbol{G}}_d$ conditioned on the FLAME expression parameters ψ and pose parameters ϕ. The refined mesh $\mathcal{T}_r$ is defined as follows:

$$\begin{aligned} \mathcal{T}_r(\beta, \psi, \phi) =& \{\mathcal{V}_r(\beta, \psi, \phi), \mathcal{F}\}, \\ \text{where } \mathcal{V}_r(\beta, \psi, \phi) =& \mathcal{V}(\beta, \psi, \phi) + \mathcal{S}(\hat{G}_d). \end{aligned} \tag{2}$$

$\mathcal{S}(\cdot)$ samples values based on the UV coordinates.

In contrast to previous geometry refinement networks [13] that rely on MLPs to predict per-vertex offsets, our approach uses a UV displacement map, which inherently promotes smoothness in the refined mesh due to the locality properties of Convolutional Neural Networks (CNNs). Additionally, by using $\mathcal{S}(\cdot)$, our geometry refinement supports unlimited mesh resolution, i.e., the computation cost does not increase as the number of vertices increases.

Disentangled Neural Texture. Given the strengths of neural textures in expressing high-quality dynamic textures and rendering efficiency [28], we adopt deferred neural rendering [38] to generate colors for facial regions. To model observations more reasonably, we disentangle neural textures $\hat{\boldsymbol{T}} \in \mathbb{R}^{1024\times 1024\times 4}$ into three components:

$$\hat{\boldsymbol{T}} = \hat{\boldsymbol{T}}_{di} + \hat{\boldsymbol{T}}_v + \hat{\boldsymbol{T}}_{dy}, \text{where } \hat{\boldsymbol{T}}_{[\,]} \in \mathbb{R}^{1024\times 1024\times 4}. \tag{3}$$

The diffuse texture $\hat{\boldsymbol{T}}_{di}$ is defined as learnable parameters, representing the base diffuse colors of each face. The view texture $\hat{\boldsymbol{T}}_v$ and dynamic texture $\hat{\boldsymbol{T}}_{dy}$ are predicted using CNNs conditioned on the view vector $\boldsymbol{d}$ and FLAME expression parameters ψ respectively to capture view-dependent effects and dynamic texture details.

Per-Pixel Texture Decoding. To achieve fast and high-fidelity rendering, we utilize a compact MLP with just 307 learnable parameters for per-pixel decoding [28] to produce RGB colors. Unlike PiCA [28], our RGB colors are predicted solely from UV coordinates and neural textures, which enhances generalization to unseen expressions. Excluding the XYZ coordinate inputs from the decoder prevents from overfitting to a specific coordinate system, thereby improving the renderings for novel expressions.

3.2. Wearable Gaussian Hair

We adopt 3DGS [19] for hair modeling since it can better reconstruct high-frequency volumetric structures than mesh-

based representations [13, 28]. Specifically, we first select one training frame (all views) to build a 3DGS-based canonical human hair with static modeling. For dynamic modeling, a rigid transformation is computed via the ICP algorithm [3] to align the canonical hair with each new frame. Additionally, an MLP-based deformation field [7, 33, 53] accounts for subtle non-rigid movements.

Preliminaries: 3D Gaussian Splatting. Given calibrated multi-view images and an initial point cloud (e.g., from SfM [35]), a *static* scene can be reconstructed using a set of anisotropic Gaussians $\mathcal{G} = \{x^i, r^i, s^i, o^i, sh^i\}_{i=1:N}$ [19]. Here, i represents the i-th Gaussian, N the number of Gaussians, $x^i \in \mathbb{R}^3$ the center of the i-th Gaussian, $r^i \in \mathbb{R}^4$ the orientation (represented by a unit quaternion), $s^i \in \mathbb{R}^3$ the scale, $o^i \in \mathbb{R}$ the opacity, and $sh^i \in \mathbb{R}^{48}$ the spherical harmonics coefficients (up to degree 3), used to model view-dependent appearance.

To render a pixel's color $\boldsymbol{C}$, all 3d Gaussians intersected with its view vector $\boldsymbol{d}$ are blended using alpha blending:

$$\boldsymbol{C} = \sum_i c_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j), \tag{4}$$

where c_i is the color of the i-th Gaussian computed from sh^i and the view vector $\boldsymbol{d}$. The blending weight α'_i is given by evaluating the 2D projection of the i-th Gaussian [55] multiplied by o^i. All Gaussians are sorted by depth before performing the alpha blending calculation.

Static Modeling of the Canonical Hair. To obtain the canonical human hair $\mathcal{G}_c = \{x^i_c, r^i_c, s^i_c, o^i_c, sh^i_c\}_{i=1:N}$, we optimize a 3DGS from multi-view images of one chosen frame. Note that we initialize the point cloud by sampling on- and off-surface points according to the scalp region of the tracked FLAME mesh and only use image pixels under the hair mask regions for photometric training.

Rigid Hair Transformation between Two Frames. To handle head movement between different frames, we compute per-frame rigid transformations $\{R_i, t_i\}_{i=1:N_f}$ relative to the FLAME mesh in the canonical frame using the ICP algorithm [3]:

$$(R_i, t_i) = \text{ICP}(\mathcal{V}^{scalp}(\beta_i, \psi_i, \phi_i), \mathcal{V}^{scalp}(\beta_c, \psi_c, \phi_c)), \tag{5}$$

where N_f represents the total number of training frames, β_c, ψ_c, and ϕ_c the FLAME parameters of the canonical frame, $\mathcal{V}^{scalp}$ the pre-defined scalp vertices. ICP$(\cdot)$ computes an alignment (i.e., a rigid transformation) by minimizing the Euclidean distance between the two point sets.

With the rigid transformations, we obtain initial transformed hair Gaussians $\mathcal{G}_r = \{x^i_r, r^i_r, s^i_c, o^i_c, sh^i_c\}_{i=1:N}$ which are used for the next dynamic hair modeling.

Non-Rigid Hair Deformation between Two Frames. To account for variations caused by different poses/expressions and achieve sharper renderings, we learn a non-rigid deformation field parameterized by an MLP $\mathcal{M}_d$:

$$\mathcal{M}_d : \psi \rightarrow (\delta x, \delta r, \delta s, \delta o, \delta sh), \tag{6}$$

where ψ represents the FLAME expression parameters. The final Gaussian hair including both rigid and non-rigid deformations is $\mathcal{G}_d = \{x^i_r + \delta x^i, r^i_r + \delta r^i, s^i_c + \delta s^i, o^i_c + \delta o^i, sh^i_c + \delta sh^i\}_{i=1:N}$.

3.3. Occlusion-Aware Blending

A basic idea for blending is to compare the depth maps of the 3DGS hair and facial mesh, and set the color of the final image pixel to that of the closer one (i.e., hard-blending). In practice, our occlusion-aware blending module (Fig. 2) needs to solve two critical challenges: training stability and blending artifacts (e.g., hairline seams).

Ensuring Stable Training. We adopt a simpler but more robust "near-z" depth $\hat{\boldsymbol{D}}_{nz}$ for our occlusion test, which is defined as the depth value of the first Gaussian (depth sorted) whose opacity value is larger than a predefined threshold (0.05 in our settings). If an image pixel's "near-z" depth is larger than its mesh depth, we know the GS hair is occluded by the facial mesh with high confidence. In contrast, using 3DGS-rendered depth for occlusion test is unstable because the rendered GS hair depth is close to mesh depth, which fluctuates due to minor training errors and causing a frequently changing occlusion state. We denote the resulting binary occlusion mask as $\boldsymbol{M}_o = \hat{\boldsymbol{D}}_{nz} < \hat{\boldsymbol{D}}_h$.

Reducing Blending Artifacts. Firstly, we propose an early-stopping rendering strategy. Specifically, for regions under $\boldsymbol{M}_o$, there exist Gaussians before the mesh, which should be accounted for during rendering, and Gaussians occluded by the mesh, which should be ignored during rendering. Thus, during the Gaussian rendering process, we will stop the color/alpha accumulation if the next Gaussian (depth sorted) is too far (i.e., the other side of the head) from the current one for a given ray, obtaining an accumulated alpha map $\boldsymbol{A}_g$ of the Gaussian hair for later blending.

Then, to further reduce artifacts around blending boundaries (especially the hairline) in the final renderings, we apply the Gaussian smoothing [8] to the binary occlusion mask $\boldsymbol{M}_o$, resulting in a soft-edge occlusion mask $G(\boldsymbol{M}_o)$.

The final blending map for the hair is computed as $\hat{\boldsymbol{A}}_{hair} = \boldsymbol{A}_g \cdot G(\boldsymbol{M}_o)$, and the final rendering $\hat{\boldsymbol{I}}$ of our hybrid representation is then given by:

$$\hat{\boldsymbol{I}} = \hat{\boldsymbol{A}}_{hair} \cdot \hat{\boldsymbol{I}}_{hair} + (\boldsymbol{1} - \hat{\boldsymbol{A}}_{hair}) \cdot \hat{\boldsymbol{I}}_{head}. \tag{7}$$

4. Optimizing Head Avatars

Directly optimizing a complete hybrid facial mesh and Gaussian hair avatar from scratch is highly under-constrained and thus inherently unstable. To address this, our optimization process for MeGA is divided into three sequential stages, including facial mesh optimization, canonical hair optimization, and joint optimization.

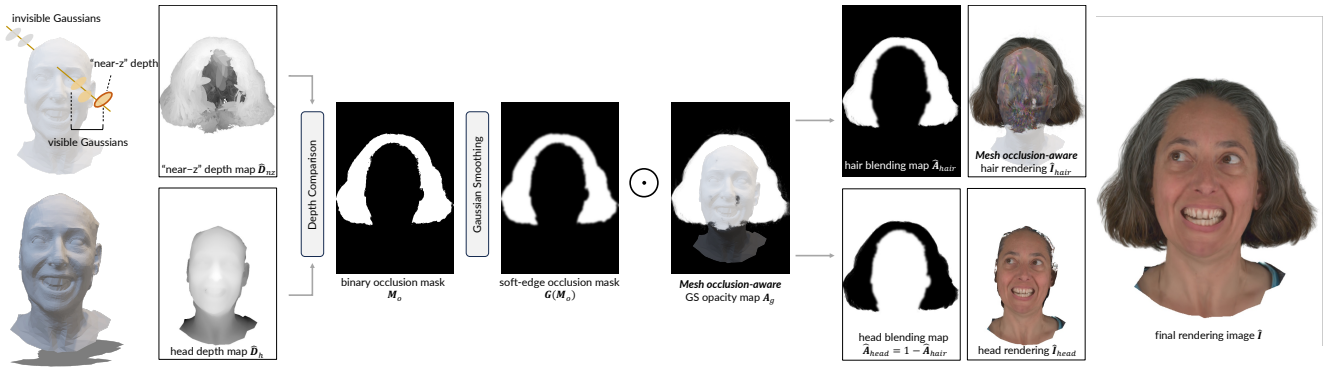

Figure 2. **Mesh Occlusion-Aware Blending.** By comparing the hair "near-z" depth map $\hat{\boldsymbol{D}}_{nz}$ and the head depth map $\hat{\boldsymbol{D}}_h$, we find pixels that should use hair renderings (white regions in $\boldsymbol{M}_o$). Further combining soft-edge occlusion mask $G(\boldsymbol{M}_o)$ with *mesh occlusion-aware* hair opacity map $\boldsymbol{A}_g$ which only account for visible Gaussians (i.e., in front of the mesh), we obtain the blending map for final renderings.

Learnable parameters. For clarity, we list all learnable parameters here. For the Gaussian hair, θ_{gs} refers to all learnable parameters of the canonical Gaussian hair, while θ_{def} represents the MLP parameters for the hair deformation field. For the facial mesh, $\hat{\boldsymbol{T}}_{di}$ is a learnable latent map (i.e., neural texture [38]) for representing diffuse color. θ_v refers to the parameters of the view texture decoder, from which a view-dependent latent map $\hat{\boldsymbol{T}}_v$ is produced. θ_{dy} refers to the parameters of the dynamic texture decoder, from which an expression-dependent latent map $\hat{\boldsymbol{T}}_{dy}$ is produced. θ_{disp} represents the parameters of the geometry decoder, from which a UV displacement map $\hat{\boldsymbol{G}}_d$ is produced. θ_{pix} specifies the parameters of the pixel decoder $\mathcal{D}_{pix}$ that decodes neural textures into RGB colors.

Optimizing Facial Mesh. In the first stage, we optimize all learnable parameters related to the facial mesh (i.e., $\hat{\boldsymbol{T}}_{di}, \theta_v, \theta_{dy}, \theta_{disp}$, and θ_{pix}) with two per-pixel photometric losses $\mathcal{L}^F_{pho}$ and $\mathcal{L}^F_{di\text{-}pho}$, a D-SSIM loss $\mathcal{L}^F_{ssim}$, a shrink loss $\mathcal{L}^F_{shr}$, two depth-based losses $\mathcal{L}^F_d$ and $\mathcal{L}^F_n$, and three regularization losses $\mathcal{L}^F_{lap}$, $\mathcal{L}^F_{nc}$, and $\mathcal{L}^F_{el}$.

Photometric losses. $\mathcal{L}^F_{pho}$ and $\mathcal{L}^F_{ssim}$ provide supervisions for rendered facial colors as:

$$\begin{aligned} \mathcal{L}^F_{pho} &= ||\boldsymbol{I}_{head} - \hat{\boldsymbol{I}}_{head}||_2, \\ \mathcal{L}^F_{ssim} &= 1 - \text{SSIM}(\boldsymbol{I}_{head}, \hat{\boldsymbol{I}}_{head}), \end{aligned} \tag{8}$$

where $\boldsymbol{I}_{head}$ is the ground truth image of the head part.

We introduce an extra L2-based photometric loss $\mathcal{L}^F_{di\text{-}pho} = ||\boldsymbol{I}_{head} - \hat{\boldsymbol{I}}^{di}_{head}||_2$ to promote more meaningful texture decomposition, where $\hat{\boldsymbol{I}}^{di}_{head}$ is decoded from only the diffuse latent textures $\hat{\boldsymbol{T}}_{di}$.

Geometric losses. We use depth and screen-space normal losses to refine the geometry of the facial mesh as follows:

$$\begin{aligned} \mathcal{L}^F_d &= ||(\boldsymbol{D}_h - \hat{\boldsymbol{D}}_h) \odot \boldsymbol{M}_d||_1, \\ \mathcal{L}^F_n &= ||N(\boldsymbol{D}_h) - N(\hat{\boldsymbol{D}}_h) \odot \boldsymbol{M}_d||, \end{aligned} \tag{9}$$

where $\boldsymbol{D}_h$ denotes the depth map derived from multi-view images using Metashape software [31]. $\hat{\boldsymbol{D}}_h$ is the depth map rasterized by our facial mesh and $N(\cdot)$ calculates screen space normals [28]. $\boldsymbol{M}_d$ is used to penalize those pixels whose depth errors are less than a depth threshold δ_D (set to 5mm), minimizing the effect of noise.

Shrink loss. To address the issue of the FLAME scalp often being oversized and overlapping with the hair, we introduce a shrink regularization loss $\mathcal{L}^F_{shr}$ for the scalp vertices,

$$\mathcal{L}^F_{shr} = ||\mathcal{V}^{scalp}_r(\beta, \psi, \phi) - \text{Mean}(\mathcal{V}^{scalp}(\beta, \psi, \phi))||_2, \tag{10}$$

where $\mathcal{V}^{scalp}_r$ are the scalp vertices visible in the current frame and are obtained by projecting hair masks back to the deformed FLAME mesh. By shrinking the scalp towards a fixed center, the Gaussians can be optimized to their correct locations without being obscured by a wrong scalp mesh.

Regularizations. Three regularization losses are used to ensure a reasonable facial mesh (e.g., no face crossing, reversing). The mesh Laplacian loss $\mathcal{L}^F_{lap}$ and normal consistency loss $\mathcal{L}^F_{nc}$ smooth the facial mesh, while the edge length loss $\mathcal{L}^F_{el}$ keeps the rigidity of the mesh as much as possible.

In summary, the complete training loss for our facial mesh is formulated as a weighted sum of these loss terms:

$$\begin{aligned} \mathcal{L}^F = \lambda_p \mathcal{L}^F_{pho} + 3 \cdot \lambda_p \mathcal{L}^F_{di\text{-}pho} + \lambda_d \mathcal{L}^F_d \\ + \lambda_n \mathcal{L}^F_n + \lambda_{ss} \mathcal{L}^F_{ssim} + \lambda_{sh} \mathcal{L}^F_{shr} + \mathcal{L}^F_{reg}, \end{aligned} \tag{11}$$

where $\mathcal{L}^F_{reg} = \lambda_{lap} \mathcal{L}^F_{lap} + \lambda_{nc} \mathcal{L}^F_{nc} + \lambda_{el} \mathcal{L}^F_{el}$.

Optimizing Canonical Gaussian Hair. Following the initialization with points sampled around the scalp mesh (as mentioned in Sec. 3.2), we optimize the canonical Gaussian hair parameters (i.e., θ_{gs}) using two appearance losses $\mathcal{L}^H_{pho}$ and $\mathcal{L}^H_{ssim}$ as in 3DGS [19], a silhouette loss $\mathcal{L}^H_{silh}$, and a regularization loss $\mathcal{L}^H_{sol}$.

Specifically, two appearance losses are defined as:

$$\begin{aligned} \mathcal{L}^H_{pho} &= ||\boldsymbol{I}_{hair} - \hat{\boldsymbol{I}}_{hair}||_2, \\ \mathcal{L}^H_{ssim} &= 1 - \text{SSIM}(\boldsymbol{I}_{hair}, \hat{\boldsymbol{I}}_{hair}), \end{aligned} \tag{12}$$

where $\boldsymbol{I}_{hair}$ is the ground truth image of the hair part.

To encourage better disentanglement between the facial

mesh and Gaussian hair, we introduce a silhouette loss:

$$\mathcal{L}_{silh}^{H} = ||(\boldsymbol{M}_{hair} - \hat{\boldsymbol{A}}_{hair}) \odot \boldsymbol{\Delta}||_1, \quad \text{where } \boldsymbol{\Delta}(x_i) = \min_{x_j \in \boldsymbol{M}_{hair}} (||x_i - x_j||_2). \tag{13}$$

where $\boldsymbol{M}_{hair}$ is the ground truth hair mask, obtained using a standard facial parsing algorithm [21]. $\boldsymbol{\Delta}(\cdot)$ is a weighting function that ensures distant incorrect pixels in the rendered mask are penalized more heavily than pixels that are closer.

We also introduce a regularization loss that encourages the Gaussian hair to generate a solid hair mask, except for its boundary regions. Mathematically, this loss is defined as: $\mathcal{L}_{sol}^{c} = ||(\mathbf{1} - \hat{\boldsymbol{A}}_{hair}) \odot \text{Erode}(\boldsymbol{M}_{hair})||_1$, where $\text{Erode}(\cdot)$ represents the erosion operation.

In summary, the complete loss used to train our canonical hair is defined as:

$$\mathcal{L}^{H} = \lambda_p \mathcal{L}_{pho}^{H} + \lambda_{ss} \mathcal{L}_{ssim}^{H} + \lambda_{sil} \mathcal{L}_{silh}^{H} + \lambda_{sol} \mathcal{L}_{sol}^{H}. \tag{14}$$

Joint Optimization. With proper initializations of the neural mesh and canonical 3DGS hair, we jointly optimize the hybrid mesh-Gaussian avatar across all frames, with a primary focus on improving the quality of the face-hair overlapping region. The objective function is defined as:

$$\mathcal{L} = \lambda_p \mathcal{L}_{pho} + 3 \cdot \lambda_p \mathcal{L}_{di \cdot pho}^{F} + \lambda_{ss} \mathcal{L}_{ssim} + \lambda_{sol} \mathcal{L}_{sol}^{H} + \lambda_n (||\delta r||_2 + ||\delta s||_2 + ||\delta o||_2 + ||\delta c||_2) + \lambda_a \mathcal{L}_{aiap}. \tag{15}$$

In this stage, we optimize for $\theta_{def}, \hat{\boldsymbol{T}}_{di}, \theta_v$, and θ_{dy}. Additionally, we introduce new regularizations to constrain the per-Gaussian update and an as-isometric-as-possible loss $\mathcal{L}_{aiap}$ [33] to encourage the rigidity of the Gaussian hair.

5. Editing Head Avatars

Due to the disentangled facial mesh and Gaussian hair, our MeGA naturally facilitates various editing operations.

Hairstyle Alteration. As shown in Fig. 1, our approach can easily update A's hairstyle with B's after alignment (with scaling). Specifically, we load subject A's facial mesh (i.e., $\hat{\boldsymbol{T}}_{di}, \theta_v, \theta_{dy}, \theta_{disp}$, and θ_{pix}) and load subject B's Gaussian hair (i.e., θ_{gs} and θ_{def}). Then, an ICP-based alignment (with scaling) is conducted to align B's hair to A's.

Facial Texture Editing. Our MeGA can easily support texture editing by updating the diffuse neural texture map $\hat{\boldsymbol{T}}_{di}$ according to the painted image $\boldsymbol{I}_p$ and its corresponding mask $\boldsymbol{M}_p$ similar to NeuMesh [48]. Specifically, to edit facial textures, we first remap the 2d painting mask to the UV space, obtaining a mask $\boldsymbol{M}_p^{uv}$. Only the latent codes under this mask are optimized during the subsequent optimization process. Then we optimize these codes in the diffuse texture map $\hat{\boldsymbol{T}}_{di}$ with a learning rate 0.01 and the pixel decoder θ_{pix} with a learning rate 0.0001. Slightly finetuning the pixel decoder allows it to show new colors that are not seen during training head avatars.

Note that we calculate losses for the complete image on the view $\boldsymbol{I}_p$ and calculate losses outside the painting mask on other views. Optimizing the losses on other views serves as a regularization of the pixel decoder $\mathcal{D}_{pix}$, ensuring minimal changes on the non-painting regions.

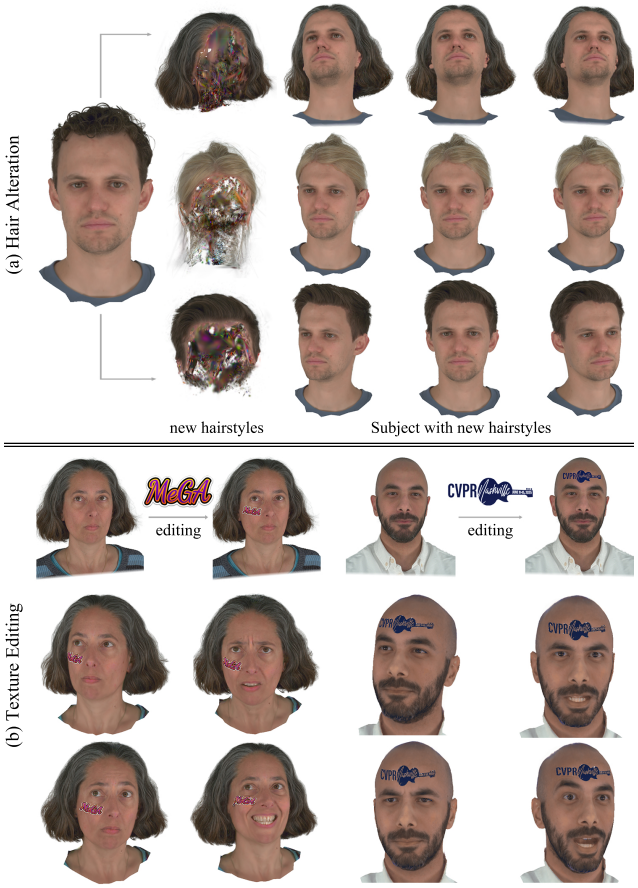

Figure 3. **Hairstyle Alteration and Texture Editing.** MeGA naturally supports hairstyle alteration and texture editing. The edited head avatar can be rendered in novel views and expressions.

6. Experiments

We evaluate our approach on the NeRSemble dataset [20], which contains multi-view videos of each subject and calibrated camera parameters of all 16 cameras. GaussianAvatars [32] downsample the images to a resolution of 802×550 and generate a foreground mask for each image. Based on their processed images, we further obtain facial parsing results for each image using an open-source algorithm [21] and depth maps for each frame using Metashape software [31].

We train our MeGA using the same train/test splits as GaussianAvatars [32]. Specifically, 9 out of 10 expression sequences and 15 out of 16 available cameras are used for training, while the remaining camera and expression sequence are reserved for evaluation. All metrics are calculated based on image pixels under the rasterization mask. The facial geometry is evaluated using the Mean Absolute Error (MAE) between the reconstructed depth maps and our

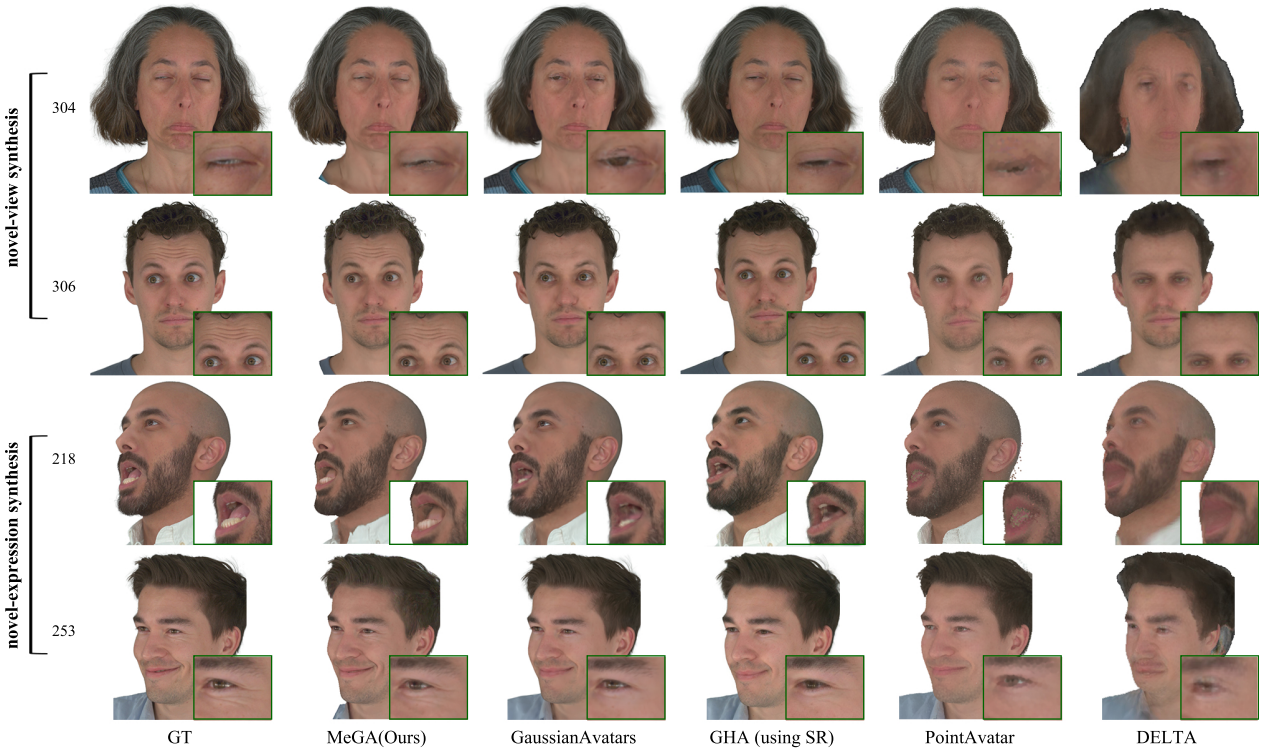

Figure 4. **Comparisons with State-of-the-Art Methods.** MeGA generates more realistic facial renditions compared to previous state-of-the-art methods, especially in terms of detailed skin textures. Note that Gaussian Head Avatar (GHA) uses a super-resolution (SR) module.

Table 1. **Comparisons with State-of-the-Art Methods.** MeGA achieves better LPIPS, SSIM, and PSNR (1dB higher than the 2nd best method). We bold (underline) the best (2nd best) results.

Method	Novel-View Synthesis			Novel-Expr. Synthesis		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
DELTA	24.62	0.871	0.138	22.60	0.858	0.158
PointAvatar	27.08	0.918	0.091	25.79	0.916	0.103
Gaussian Head Avatar	29.48	0.894	0.084	22.02	0.847	0.156
GaussianAvatars	33.54	0.951	0.055	31.45	0.947	0.060
MeGA(Ours)	**34.11**	**0.954**	**0.052**	**32.59**	**0.949**	**0.057**

rasterized depth maps.

6.1. Comparisons with State-of-the-Art Methods

We conduct comparisons with GaussianAvatars [32], Gaussian Head Avatar with super-resolution (SR) [47], PointAvatars [52], and DELTA [10] to demonstrate the superiority of our method. All baselines are trained from scratch using their public codes and the training details are provided in our **Supp. Mat.** Tab. 1 shows that our approach achieves the best PSNR, SSIM, and LPIPS averaged among all 9 subjects. As shown in Fig. 4, due to the use of expression-dependent dynamic textures (i.e., $\hat{\boldsymbol{T}}_{dy}$), our MeGA can model more subtle geometric details (e.g., wrinkles in subject 306 and 253). Besides, due to the integral-based rendering, 3DGS-based facial rendering tends to produce blurry or interpenetrated results around the eye and mouth regions (e.g., subject 304 and 218). The possible reason is that when fitting a close-eye expression, the pixels around the eye region should ideally only use the Gaussians of the eyelids for rendering. However, the 3DGS rendering process cannot distinguish between the Gaussians of the eyelids and eyeballs, thereby using both of them to perform rendering and producing interpenetrated artifacts. The blurry mouth is caused by a similar reason. Note that while Gaussian Head Avatar produces promising results for novel-view synthesis, it struggles with novel expression rendering due to its heavy reliance on the implicit deformation and super-resolution module. More results are shown in our **Supp. Mat.**

6.2. Experiments on Head Editing

We present our results for qualitative evaluation only, as, to the best of our knowledge, no previous methods[1] are suitable for the following types of head editing.

MeGA supports changing someone's hairstyle to a new one (i.e., short, medium, and long hair) from another MeGA-pretrained model (Fig. 3a). Recomposed head avatars can be rendered in novel views and expressions. Fig. 3b demonstrates the texture editing functionality. Given a painted image of the subject and the corresponding painting mask, MeGA can embed this modification into the 3D head avatar and render view-consistent images in novel views and expressions.

[1]Previous mesh-based methods [10, 13, 28] are not suitable for texture editing due to the entanglement of base colors and view-dependent effects. DELTA [10] produces rather poor renderings in our settings.

Table 2. **Ablation Studies on Subject 306.** We demonstrate the effectiveness of each proposed component. (a) shows the importance of our disentangled texture maps in generating high-quality renderings. (b) shows the positive effects of our mesh subdivision and UV displacement map. (c) shows the superiority of our occlusion-aware blending. A blank entry indicates the same settings as "MeGA (Ours)".

Label	Name	Texture	Mesh Geom.	Blending	PSNR ↑	SSIM ↑	LPIPS ↓	Geo. MAE ↓
	MeGA (Ours)				**33.57**	**0.963**	**0.040**	**2.25mm**
(a.1)	MeGA-noview	w/o $\hat{T}_v$			31.68	0.958	0.053	2.87mm
(a.2)	MeGA-nodyn	w/o $\hat{T}_{dy}$			32.81	0.959	0.050	2.91mm
(b.1)	MeGA-nosubdiv		no subdivision		32.81	0.962	0.046	3.38mm
(b.2)	MeGA-nodisp		no $\hat{\boldsymbol{G}}_d$		32.99	0.959	0.054	7.48mm
(c.1)	MeGA-gsdepth			using alpha-acc. depths	27.94	0.950	0.068	**2.25mm**
(c.2)	MeGA-allGS			no early-stopping	31.42	0.957	0.048	**2.25mm**

6.3. Ablation Studies

In this section, we present a series of ablation studies to verify the effectiveness of our major design choices.

Disentangled Texture Maps. Tab. 2 (a.1) and (a.2) illustrate the roles of our two disentangled texture maps, with the corresponding visual results shown in Fig. 5. When the view texture $\hat{\boldsymbol{T}}_v$ is disabled (MeGA-noview), MeGA struggles to handle view-dependent effects and fails to capture highlights in the eyes. When the expression-dependent dynamic texture $\hat{\boldsymbol{T}}_{dy}$ is disabled (MeGA-nodyn), MeGA loses the ability to model detailed skin appearance (e.g., the forehead wrinkles). Disabling any of them results in worse quantitative metrics (31.68/32.81 vs. 33.57).

Mesh Geometry. We investigate the effect of mesh subdivision and the use of the UV displacement map $\hat{\boldsymbol{G}}_d$ for enhancing geometry details. The quantitative results are reported in Tab. 2 (b.1) and (b.2). Without mesh subdivision, only 5023 vertices are adapted to fit the facial depths, leading to inferior facial geometry and renderings (3.38mm vs. 2.25mm Geo. MAE, 32.81 vs. 33.57 PSNR). Using a UV displacement map $\hat{\boldsymbol{G}}_d$ significantly improves the evaluation metrics (2.25mm vs. 7.48mm Geo. MAE, 33.57 vs 32.99 PSNR). The visual results are shown in Fig. 5.

Blending Strategies. To verify the effectiveness of our mesh occlusion-aware blending approach, we test alternative blending strategies and report the quantitative results in Tab. 2 (c.1)-(c.2). "MeGA-gsdepth" attempts to obtain the visibility of the Gaussian hair using 3DGS-rendered depths, instead of the "near-z" depths. However, 3DGS-rendered depths may fluctuate due to minor training errors and make occlusion relations between the head and 3DGS hair changing constantly, resulting in the optimization objective of 3DGS shifting throughout the training process and unstable optimization. "MeGA-allGS" disables our early-stopping strategy and uses both invisible and visible Gaussians for hair rendering. In this case, if a single Gaussian mistakenly appears in front of the facial mesh, the invisible Gaussians will be used to fit the facial appearance, disrupting the learning of facial textures and leading to inferior facial renderings (31.42 vs. 33.57 PSNR and Fig. 5).

Figure 5. **Ablation Studies** on Disentangles Texture Maps, Geometry Refinement, and Blending Strategies. Disabling the view texture $\hat{T}_v$ and dynamic texture $\hat{T}_{dy}$ loses the highlights in the eyes and the forehead wrinkles, respectively. Removing any component in our geometry refinement and occlusion-aware blending module degrades the final renderings.

Loss Functions. Removing any loss function degrades the performance. More details are shown in our **Supp. Mat.**

7. Conclusion

In this paper, we present hybrid mesh-Gaussian head avatars (MeGA), which employ neural mesh for face modeling and 3DGS for hair modeling. For high-quality facial modeling, we enhance the FLAME mesh and decode a UV displacement map for personalized geometric details. Facial colors are decoded via a lightweight MLP from a neural texture map that consists of disentangled diffuse texture $\hat{\boldsymbol{T}}_{di}$, view-dependent texture $\hat{\boldsymbol{T}}_v$, and dynamic texture $\hat{\boldsymbol{T}}_{dy}$. For high-quality hair modeling, we build a static 3DGS hair and employ a rigid transformation combined with an MLP-based deformation field for animation. The final renderings are obtained by blending the hair and head parts with our occlusion-aware blending module. In addition to achieving the best rendering results, MeGA naturally supports various editing functionalities, including hairstyle alteration and texture editing.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2023YFF0905104), the Natural Science Foundation of China (No. 62361146854) and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Ziqian Bai, Feitong Tan, Zeng Huang, Kripasindhu Sarkar, Danhang Tang, Di Qiu, Abhimitra Meka, Ruofei Du, Mingsong Dou, Sergio Orts-Escolano, Rohit Pandey, Ping Tan, Thabo Beeler, Sean Fanello, and Yinda Zhang. Learning personalized high quality volumetric head avatars from monocular RGB videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 16890–16900. IEEE, 2023. 1

[2] Linchao Bao, Xiangkai Lin, Yajing Chen, Haoxian Zhang, Sheng Wang, Xuefei Zhe, Di Kang, Haozhi Huang, Xinwei Jiang, Jue Wang, et al. High-fidelity 3d digital human head creation from rgb-d selfies. *ACM TOG*, 41(1):1–21, 2021. 2

[3] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14 (2):239–256, 1992. 4

[4] Shrisha Bharadwaj, Yufeng Zheng, Otmar Hilliges, Michael J. Black, and Victoria Fernández Abrevaya. FLARE: fast learning of animatable and relightable mesh avatars. *ACM Trans. Graph.*, 42(6):204:1–204:15, 2023. 2

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, pages 187–194. ACM, 1999. 2

[6] Chuhan Chen, Matthew O'Toole, Gaurav Bharaj, and Pablo Garrido. Implicit neural head synthesis via controllable local deformation fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 416–426. IEEE, 2023. 2

[7] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Monogaussianavatar: Monocular gaussian point-based head avatar. In *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024- 1 August 2024*, page 58. ACM, 2024. 4

[8] Moo K. Chung. Gaussian kernel smoothing. *CoRR*, abs/2007.09539, 2020. 4

[9] Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Headgas: Real-time animatable head avatars via 3d gaussian splatting. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part II*, pages 459–476. Springer, 2024. 1, 2

[10] Yao Feng, Weiyang Liu, Timo Bolkart, Jinlong Yang, Marc Pollefeys, and Michael J. Black. Learning disentangled avatars with hybrid 3d representations. *CoRR*, abs/2309.06441, 2023. 2, 7

[11] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, pages 8649–8658, 2021. 2

[12] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM TOG*, 41(6):1–12, 2022. 2

[13] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular RGB videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18632–18643. IEEE, 2022. 1, 2, 3, 4, 7

[14] Zhenyi He, Ruofei Du, and Ken Perlin. Collabovr: A reconfigurable framework for creative collaboration in virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2020, Recife/Porto de Galinhas, Brazil, November 9-13, 2020*, pages 542–554, Brazil, 2020. IEEE. 1

[15] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A realtime nerf-based parametric head model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 20342–20352. IEEE, 2022. 1

[16] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *CVPR*, pages 20374–20384, 2022. 2

[17] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM TOG*, 36(6):1–14, 2017. 2

[18] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM TOG*, 34(4):1–14, 2015. 2

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023. 1, 3, 4, 5

[20] Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. Nersemble: Multi-view radiance field reconstruction of human heads. *ACM TOG*, 42(4): 161:1–161:14, 2023. 6

[21] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[22] Jiameng Li, Yue Shi, Jiezhang Cao, Bingbing Ni, Wenjun Zhang, Kai Zhang, and Luc Van Gool. Mipmap-gs: Let gaussians deform with scale-specific mipmap for anti-aliasing rendering. *CoRR*, abs/2408.06286, 2024. 1

[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 Trans. Graph.*, 36(6):194:1–194:17, 2017. 1, 2

[24] Stephen Lombardi, Jason M. Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Trans. Graph.*, 37(4):68, 2018. 1, 2

[25] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM TOG*, 38(4):1–14, 2019. 2

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

[27] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhöfer, Yaser Sheikh, and Jason M. Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Trans. Graph.*, 40(4):59:1–59:13, 2021. 1

[28] Shugao Ma, Tomas Simon, Jason M. Saragih, Dawei Wang, Yuecheng Li, Fernando De la Torre, and Yaser Sheikh. Pixel codec avatars. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 64–73. Computer Vision Foundation / IEEE, 2021. 1, 2, 3, 4, 5, 7

[29] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers, SIGGRAPH 2024, Denver, CO, USA, 27 July 2024- 1 August 2024*, page 60. ACM, 2024. 2

[30] Sergio Orts-Escolano, Christoph Rhemann, Sean Ryan Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles T. Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien P. C. Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST 2016, Tokyo, Japan, October 16-19, 2016*, pages 741–754, ., 2016. ACM. 1

[31] Jin-Si R Over, Andrew C Ritchie, Christine J Kranenburg, Jenna A Brown, Daniel D Buscombe, Tom Noble, Christopher R Sherwood, Jonathan A Warrick, and Phillipe A Wernette. Processing coastal imagery with agisoft metashape professional edition, version 1.6—structure from motion workflow documentation. Technical report, US Geological Survey, 2021. 5, 6

[32] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 20299–20309. IEEE, 2024. 1, 2, 6, 7

[33] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 5020–5030. IEEE, 2024. 4, 6

[34] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 130–141. IEEE, 2024. 2

[35] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4104–4113. IEEE Computer Society, 2016. 4

[36] Artem Sevastopolsky, Philip-William Grassal, Simon Giebenhain, ShahRukh Athar, Luisa Verdoliva, and Matthias Nießner. Headcraft: Modeling high-detail shape variations for animated 3dmms. *CoRR*, abs/2312.14140, 2023. 3

[37] Xiaowei Song, Jv Zheng, Shiran Yuan, Huan-ang Gao, Jingwei Zhao, Xiang He, Weihao Gu, and Hao Zhao. SA-GS: scale-adaptive gaussian splatting for training-free anti-aliasing. *CoRR*, abs/2403.19615, 2024. 1

[38] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph.*, 38(4):66:1–66:12, 2019. 1, 2, 3, 5

[39] Zach Waggoner. *My avatar, my self: Identity in video role-playing games*. McFarland, ., 2009. 1

[40] Cong Wang, Di Kang, Yan-Pei Cao, Linchao Bao, Ying Shan, and Song-Hai Zhang. Neural point-based volumetric avatar: Surface-guided neural points for efficient and photo-realistic volumetric head avatar. In *SIGGRAPH Asia 2023 Conference Papers, SA 2023, Sydney, NSW, Australia, December 12-15, 2023*, pages 50:1–50:12. ACM, 2023. 1, 2, 3

[41] Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: High-fidelity head avatars with learnable gaussian derivation, 2024. 1, 2

[42] Peng Wang, Xiaoliang Bai, Mark Billinghurst, Shusheng Zhang, Xiangyu Zhang, Shuxia Wang, Weiping He, Yuxiang Yan, and Hongyu Ji. AR/MR remote collaboration on physical tasks: A review. *Robotics Comput. Integr. Manuf.*, 72:102071, 2021. 1

[43] Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. Learning compositional radiance fields of dynamic human heads. In *CVPR*, pages 5704–5713, 2021. 2

[44] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1802–1812. IEEE, 2024. 1, 2

[45] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, pages 47:1–47:10. ACM, 2023. 1

[46] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023. 2

[47] Yuelang Xu, Bengwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1931–1941. IEEE, 2024. 2, 7

[48] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XVI*, pages 597–614. Springer, 2022. 6

[49] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 19447–19456. IEEE, 2024. 1

[50] Xiaochen Zhao, Lizhen Wang, Jingxiang Sun, Hongwen Zhang, Jinli Suo, and Yebin Liu. Havatar: High-fidelity head avatar via facial model conditioned neural radiance field. *ACM TOG*, 43(1):1–16, 2023. 2

[51] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *CVPR*, pages 13545–13555, 2022. 2

[52] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 21057–21067. IEEE, 2023. 2, 7

[53] Mingyuan Zhou, Rakib Hyder, Ziwei Xuan, and Guojun Qi. Ultravatar: A realistic animatable 3d avatar diffusion model with authenticity guided textures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1238–1248. IEEE, 2024. 4

[54] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *CVPR*, pages 4574–4584, 2023. 2

[55] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus H. Gross. EWA volume splatting. In *12th IEEE Visualization Conference, IEEE Vis 2001, San Diego, CA, USA, October 24-26, 2001, Proceedings*, pages 29–36. IEEE Computer Society, 2001. 4