

URHead: A Unified UV-Space Representation for Joint Mesh–3DGS Optimization in Head Avatars

Seonghak Lee¹, Junhee Cho¹, Jisoo Park¹, Min-Gyu Park^{2,3}, Jongmin Lee¹, Ju Hong Yoon^{2,3}, and Junseok Kwon^{†1}

¹ Chung-Ang University

² Korea Electronics Technology Institute (KETI)

³ polygom

[†] indicates corresponding author.

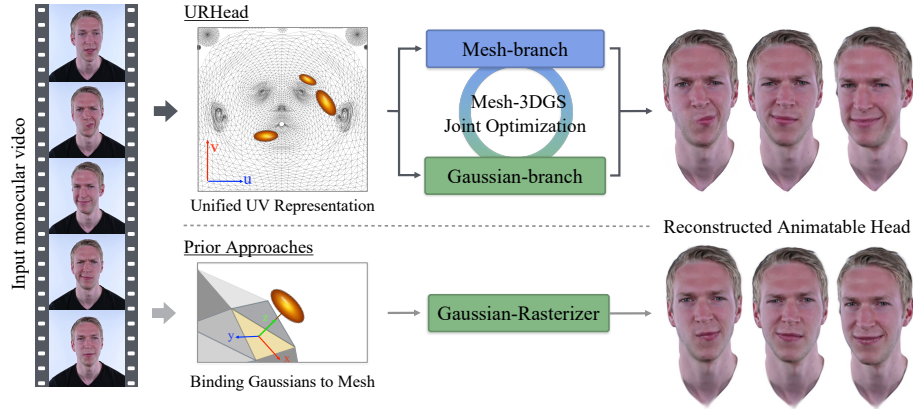


Fig. 1: Key insight behind URHead. Unlike prior approaches that bind Gaussians to mesh surfaces in 3D space (bottom), URHead introduces a unified UV-space representation that jointly parameterizes mesh and 3DGS through shared attribute maps (top). Through joint optimization over the shared UV, both representations mutually refine each other, preserving animation controllability while bridging structural coherence and visual fidelity.

Abstract. We present URHead, a unified representation for high-fidelity and animatable head avatars that fundamentally redefines mesh-Gaussian integration. While mesh-based methods offer precise geometric control but lack photorealistic detail, and Gaussian-based approaches achieve photorealism but suffer from poor structural consistency, existing hybrid solutions fail to fully leverage their complementary strengths. Our key contribution is a UV-space unification where both representations share a common UV parameterization. Through joint optimization with adaptive gaussian sampling, our method automatically learns to disentangle and allocate appropriate roles to each component. URHead maintains full parametric controllability while preserving subject-specific details, outperforms existing state-of-the-art methods in reconstruction quality and animation consistency.

Our code is available at <https://github.com/Lseonghak/URHead>

1 Introduction

Creating photorealistic digital human avatars has long been a fundamental challenge in computer graphics and vision, with applications spanning virtual reality, film production, telepresence, and gaming [5, 6, 11, 12, 14, 20, 39, 40]. Recent advances in neural rendering have dramatically improved the quality of digital humans. Nevertheless, current approaches to animatable head avatar reconstruction predominantly rely on either parametric mesh models or 3D Gaussian Splatting (3DGS). Parametric meshes provide explicit geometric structure and controllable deformation, enabling stable animation, whereas 3DGS achieves photorealistic rendering through unstructured Gaussian primitives. However, neither representation alone fully satisfies the demands of both structural coherence and high-frequency detail modeling.

Parametric facial mesh models [7, 9, 29, 34, 45] have been widely adopted as a core representation for animatable head avatars, offering explicit topology and controllable deformation through low-dimensional shape and expression parameters. This structured design ensures stable animation and semantic control. However, the same structural priors fundamentally limit geometric expressiveness: the fixed topology cannot faithfully represent high-frequency details such as wrinkles or pores, and the restricted parameter space inevitably discards subject-specific variations outside the learned basis.

In contrast, 3DGS [15] has emerged as a powerful alternative, representing scenes as anisotropic Gaussian primitives that enable real-time, high-fidelity rendering. However, Gaussians lack explicit topology, making it challenging to maintain structural coherence during animation-driven deformation. To mitigate this issue, recent hybrid approaches [13, 19, 26, 31, 38] incorporate mesh controllability by binding or initializing Gaussians on parametric surfaces. While this introduces coarse geometric guidance, the Gaussians are still optimized independently in 3D space. As a result, over-pruning in sparsely observed regions can remove necessary surface coverage and lead to hollow artifacts, ultimately undermining the structural consistency that mesh integration aims to provide.

Recent studies [2, 13, 18, 25, 31] have attempted to combine mesh and 3DGS representations, yet most adopt fundamentally decoupled formulations. Methods such as [13, 25, 31] constrain Gaussian centers to remain near mesh surfaces to inherit deformation priors, but this spatial binding restricts representational flexibility without enabling true parameter sharing. HERA [2] enhances texture fidelity on smooth surfaces, yet maintains separate mesh and Gaussian components optimized independently. The core limitation of these hybrid approaches lies in their reliance on 3D-space optimization, where mesh and Gaussians interact only through geometric proximity rather than through a shared parameterization. Consequently, feature exchange between the two remains indirect and incomplete, preventing them from fully leveraging their complementary strengths.

We present URHead, a unified representation for animatable head avatars that *jointly optimizes* mesh and 3D Gaussian primitives under a *common UV parameterization*. Unlike prior hybrid approaches that couple mesh and Gaussians only through spatial constraints in 3D space, URHead reformulates both

representations as functions defined over a shared UV domain, enabling true parameter-level integration. Our key insight is that mesh and Gaussian representations are inherently complementary: meshes provide stable low-frequency geometric structure and controllable deformation, while Gaussians excel at modeling high-frequency residual details and complex appearance variations. Existing approaches either force one representation to approximate all aspects of geometry and appearance [13, 31, 38] or combine them through decoupled optimization schemes [2, 25]. In contrast, we reformulate both representations as functions defined over a common UV parameterization, enabling true parameter-level unification instead of mere spatial binding.

Within the unified UV space, the mesh provides base geometry and appearance through shared attribute maps, while UV-parameterized Gaussians capture residual details beyond the capacity of fixed-topology meshes. This shared representation enables mesh and Gaussians to be jointly optimized, allowing coordinated refinement of structure and appearance. Furthermore, UV-space unification naturally supports error-driven adaptive Gaussian allocation. By projecting reconstruction errors onto the surface domain, Gaussians are densified in under-represented regions while redundancy is suppressed elsewhere. Unlike additive blending [19, 26] or surface-constrained spatial binding [13, 31, 38], our approach achieves integration at the representation level. Through this unified formulation, URHead preserves animation controllability while attaining photorealistic reconstruction quality, bridging structural coherence and visual fidelity. Fig. 1 illustrates our key insight. We summarize our main contributions as follows:

- **Joint UV-space optimization.** To the best of our knowledge, we are the first to jointly optimize mesh and 3D Gaussians within a shared UV-parameterized domain for animatable head avatars.
- **Unified UV representation.** We introduce a unified UV-space formulation that integrates mesh geometry and Gaussian attributes within a shared parameterized domain.
- **Error-driven Gaussian allocation.** We propose an adaptive Gaussian allocation strategy in UV space for surface-aware densification of under-represented regions.
- **Extensive evaluation.** Our approach consistently outperforms mesh-only, Gaussian-only, and prior hybrid methods on standard 3D head avatar benchmarks in reconstruction accuracy and animation stability.

2 Related Work

Mesh representation for head avatars. Mesh representations have long served as the foundation of 3D face modeling due to their explicit topology and efficient rendering. 3D Morphable Models (3DMM) pioneered statistical face modeling by learning PCA bases from 3D scans, parameterizing faces using identity and expression coefficients. Basel Face Model [28] expanded the statistical basis with higher quality scans. FaceWarehouse [3] introduced multi-linear models for identity and expression. FLAME [21] incorporated articulation

components for jaw and expression control. DECA [7] predicted FLAME parameters along with displacement maps to capture subject-specific geometric details. Deferred Neural Rendering [32] rasterizes geometry into feature maps that are decoded into view-dependent images. Deep Appearance Models [23] assign latent codes to mesh vertices and use view-conditioned MLPs to model spatially-varying BRDF. NeuTex [37] aggregates multi-view features into a consistent UV texture space via differentiable sampling, while Neural Head Avatars [10] define neural textures on the FLAME topology for avatar rendering.

Hybrid representations for head avatars. Recent advances in neural rendering have explored various hybrid representations that combine explicit and implicit modeling. NeRF [27] integrated neural fields with traditional graphics primitives. NerFace [8] combined neural radiance fields with parametric face models [1, 33] for controllable synthesis, while PointAvatar [43] used neural point clouds attached to FLAME mesh for dynamic head modeling. Implicit-explicit hybrids like NPHM [9] combined the parametric space of FLAME with neural implicit signed distance functions. IMAvatar [42] learned implicit fields on top of FLAME geometry to model detailed facial appearance. Among these hybrid approaches, 3D Gaussian Splatting [15] has emerged as a particularly powerful representation, modeling scenes as anisotropic Gaussian primitives that enable real-time rendering without neural network queries. The explicit formulation of Gaussians and their success in dynamic modeling [24, 36] have inspired their integration with parametric models, leading to Mesh-GS hybrid methods.

Mesh-3DGS representation for head avatars. Recent studies have explored hybridizing multiple 3D representations to leverage their complementary strengths in expressiveness, efficiency, and controllability. SplattingAvatar [31] embedded 3D Gaussians on mesh triangles using barycentric coordinates and normal-based displacement, deriving pose-dependent transformations from mesh deformation. GaussianAvatars [13] introduced rigged 3DGS by binding Gaussians to a FLAME mesh, enabling deformation driven by facial expression and pose parameters [21]. FlashAvatar [38] allocated Gaussians in UV space and applies expression-dependent offsets via an MLP conditioned on FLAME codes, while MonoGaussianAvatar [4] learned adaptive Gaussian transformations through deformation fields for expression-aware animation. GaussianBlendShapes [26] and RGBAvatar [19] constructed blendshapes directly in Gaussian space, enabling efficient real-time control through linear blending. While these methods anchor Gaussians to mesh geometry to inherit deformation priors, FATE [41] focused on efficient Gaussian utilization via importance-based sampling and post-processing neural baking for improved texture editability.

Positioning of our work. In contrast to prior hybrid approaches, URHead achieves parameter-level unification by jointly optimizing mesh and Gaussians within a shared UV-parameterized representation.

3 Proposed Method

The proposed URHead consists of four main components. Sec. 3.1 introduces the unified UV-space facial representation that leverages the complementary prop-

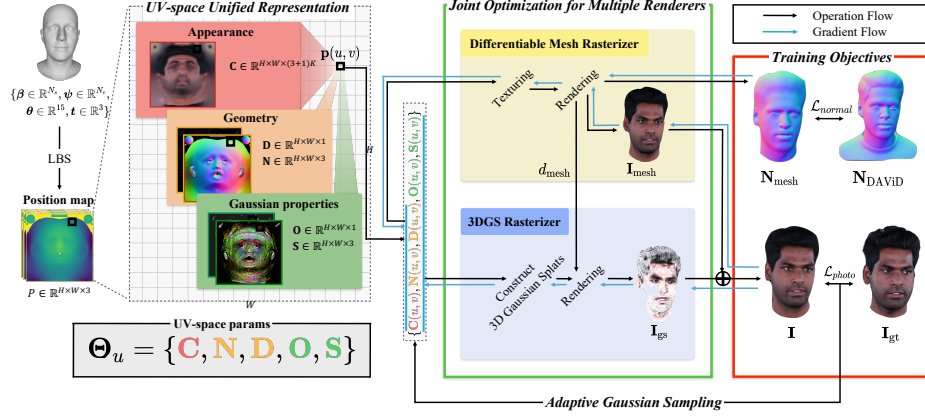


Fig. 2: Overall pipeline of URHead. Given monocular video frames and FLAME parameters (β, ψ, θ, t) , we construct a position map $\mathbf{P}$ through linear blend skinning (LBS) [21]. Our unified UV-space representation $\Theta_u = \{\mathbf{C}, \mathbf{N}, \mathbf{D}, \mathbf{O}, \mathbf{S}\}$ consists of **appearance**, **geometry**, and **gaussian-specific** parameters. The joint optimization framework employs two renderers: a differentiable mesh rasterizer takes $\{\mathbf{C}, \mathbf{N}, \mathbf{D}\}$, and produces $\mathbf{I}_{\text{mesh}}$, $\mathbf{N}_{\text{mesh}}$, and depth d_{mesh} . Meanwhile, the 3DGS rasterizer takes $\{\mathbf{C}, \mathbf{N}, \mathbf{D}, \mathbf{O}, \mathbf{S}\}$, and generates $\mathbf{I}_{\text{gs}}$ using occlusion-aware rendering. Both renderers share the UV-space parameters and are jointly optimized through gradient flows (blue arrows), enabling mutual refinement. The final image $\mathbf{I}$ is composited via alpha blending. Training objectives include photometric loss $\mathcal{L}_{\text{photo}}$ and normal consistency loss $\mathcal{L}_{\text{normal}}$ supervised by pseudo ground-truth $\mathbf{N}_{\text{DAVID}}$. The Adaptive Gaussian Sampling module allocates new points based on reconstruction error.

erties of mesh and 3DGS. Sec. 3.2 describes the joint optimization framework that aligns multiple renderers under this unified formulation. Sec. 3.3 presents an adaptive Gaussian sampling strategy for efficient UV-space allocation. Sec. 3.4 details the training objectives and overall loss function. Fig. 2 illustrates the complete URHead pipeline.

3.1 Unified UV-Parameterized Representation

Our unified UV-space representation allows structured geometry and fine-grained appearance to be jointly encoded and optimized under a common parameterization. By reformulating both mesh and Gaussian attributes in UV space, we overcome their complementary limitations: meshes are restricted by fixed topology in capturing high-frequency detail, whereas 3DGS lack structural constraints for stable deformation.

Our framework builds upon a parametric facial model [9, 21, 34] that provides a consistent UV parameterization with controllable facial expressions and poses. In our implementation, we adopt FLAME [21], though other parametric facial models could be used. FLAME defines facial parameters as $\Theta_f = \{\beta, \psi, \theta, t\}$, where $\beta \in \mathbb{R}^{N_s}$ encodes identity shape, $\psi \in \mathbb{R}^{N_e}$ models facial expressions, $\theta \in \mathbb{R}^{15}$ represents pose parameters (global rotation, neck, jaw, and eyeballs), and $t \in \mathbb{R}^3$ denotes translation. Given Θ_f , the model generates a head mesh with vertices $\mathcal{V}$ and faces $\mathcal{F}$, together with a UV parameterization that maps

each vertex to texture coordinates. This UV mapping establishes a shared parameterized domain, enabling appearance, geometry, and Gaussian attributes to be represented consistently in dense UV maps.

Leveraging the shared UV parameterization, we formulate a unified representation in a common parameterized domain:

$$\Theta_u = \{\mathbf{C}, \mathbf{N}, \mathbf{D}, \mathbf{O}, \mathbf{S}\}, \quad (1)$$

where all facial attributes are expressed in a shared UV-parameterized domain as dense maps. The representation is composed of two groups: shared attribute maps $(\mathbf{C}, \mathbf{N}, \mathbf{D})$, jointly utilized by both mesh and Gaussian renderers, and Gaussian-specific maps $(\mathbf{O}, \mathbf{S})$, which encode properties unique to Gaussian primitives. The color map $\mathbf{C} \in \mathbb{R}^{H \times W \times (3+1)K}$ stores spherical harmonics (SH) coefficients for view-dependent appearance, where $K = (d+1)^2$ with SH degree d . The normal map $\mathbf{N} \in \mathbb{R}^{H \times W \times 3}$ encodes per-pixel surface orientation in UV space, supporting both mesh shading and Gaussian rotation. The displacement map $\mathbf{D} \in \mathbb{R}^{H \times W}$ introduces learnable geometric details by displacing vertices along surface normals. These shared maps $(\mathbf{C}, \mathbf{N}, \mathbf{D})$ are jointly utilized by both the mesh and Gaussian Splatting (GS) renderers. For Gaussian rendering, we additionally define an opacity map $\mathbf{O} \in \mathbb{R}^{H \times W}$ and a scale map $\mathbf{S} \in \mathbb{R}^{H \times W \times 3}$. The opacity map controls the transparency of each Gaussian primitive, while the scale map determines its anisotropic extent along three axes. Gaussian rotation is derived from the shared normal map $\mathbf{N}$ via quaternion conversion, ensuring alignment with local surface geometry.

3.2 Joint UV-Space Optimization of Mesh and 3DGS

We jointly optimize the mesh and Gaussians to combine their complementary strengths. Optimizing them independently often leads to misaligned geometry and residual details, so a shared UV-space optimization maintains consistent updates across both representations.

The mesh renderer recovers coarse geometry and smooth appearance from the FLAME template [21], while the Gaussian renderer supplements high-frequency residual details that are difficult to capture with a template mesh alone. Given parametric inputs Θ_f , FLAME produces base vertices $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{N_v}$ for each frame. We rasterize the deformed mesh into UV space to construct a position map $\mathbf{P} \in \mathbb{R}^{H \times W \times 3}$, forming a dense 3D position field. For any UV coordinate (u, v) , the corresponding 3D position is defined as

$$\mathbf{x}(u, v) = \mathbf{P}(u, v) + \mathbf{D}(u, v) \cdot \mathbf{N}(u, v), \quad (2)$$

where $\mathbf{P}(u, v)$ denotes the base mesh position and $\mathbf{D}(u, v)$ introduces learnable displacement along the normal $\mathbf{N}(u, v)$. As $\mathbf{P}$ is updated per frame from the deformed mesh, consistent 3D–UV correspondence is preserved across renderers. **Mesh Rendering.** We obtain displaced vertices $\hat{\mathcal{V}}$ by sampling the displacement map $\mathbf{D}$ at each vertex UV coordinate (u_i, v_i) :

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \mathbf{D}(u_i, v_i) \cdot \mathbf{N}(u_i, v_i),$$

where $\mathbf{v}_i$ denotes the FLAME vertex and $\mathbf{N}$ is the shared normal map. The displaced mesh is rendered with nvdiffrast [17] via

$$\mathbf{I}_{\text{mesh}}, \mathbf{N}_{\text{mesh}}, d_{\text{mesh}} = \mathcal{R}_{\text{mesh}}(\mathbf{C}, \mathbf{N}, \hat{\mathcal{V}}, \mathcal{F}), \quad (3)$$

producing the rendered mesh image $\mathbf{I}_{\text{mesh}}$, rendered normal map $\mathbf{N}_{\text{mesh}}$, and mesh depth map d_{mesh} . The shared UV maps $\mathbf{C}$ and $\mathbf{N}$ are sampled using per-pixel UV coordinates from rasterization. To better capture geometric displacement, we refine $\mathbf{N}$ using supervision from the monocular normal estimator DAViD [30], which provides pseudo ground-truth normals. The resulting depth map d_{mesh} is later used for occlusion-aware Gaussian rendering.

Gaussian Rendering. Unlike standard 3D Gaussian Splatting [15], where Gaussian centers are directly defined in 3D space, we parameterize Gaussians in UV space as $\mathcal{G} \in \mathbb{R}^{N_{gs} \times 2}$. For each UV coordinate $(u_j, v_j) \in \mathcal{G}$, the corresponding 3D position $\mathbf{p}$ is obtained by lifting the 2D location through the position map $\mathbf{P}$. Gaussian rotation is derived from the shared normal map $\mathbf{N}$ via quaternion transformation, while opacity, scale, and color are sampled from the UV maps $\mathbf{O}$, $\mathbf{S}$, and $\mathbf{C}$, respectively.

To account for visibility interactions with the mesh, we modify the 3DGS rasterizer [15] for occlusion-aware rendering:

$$\mathbf{I}_{\text{gs}}, T = \mathcal{R}_{\text{gs}}(\mathbf{C}, \mathbf{N}, \mathbf{O}, \mathbf{S}, d_{\text{mesh}}, \mathcal{G}), \quad (4)$$

where $\mathbf{I}_{\text{gs}}$ is the rendered Gaussian image, $\mathcal{G}$ represents the lifted 3D Gaussians, and d_{mesh} is the mesh depth map. Gaussians behind the mesh surface ($z_j > d_{\text{mesh}}(x, y) + \epsilon$, with $\epsilon = 0.01$) are culled, and visible Gaussians are composited via alpha blending. The rasterizer outputs the transmittance T , representing the accumulated visibility of the Gaussian layer and regulating the contribution of the mesh layer in the final composition.

Joint Optimization of Mesh and 3DGS. Because the mesh and Gaussian renderers share the UV-space attribute maps $(\mathbf{C}, \mathbf{N}, \mathbf{D})$, gradients from both rendering paths jointly update these parameters during backpropagation. Taking the normal map $\mathbf{N}$ as an example, the total gradient decomposes as

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial \mathbf{N}} = \underbrace{\frac{\partial \mathcal{L}_{\text{photo}}}{\partial \mathbf{I}_{\text{mesh}}} \frac{\partial \mathbf{I}_{\text{mesh}}}{\partial \mathbf{N}}}_{\text{Mesh Flow}} + \underbrace{\frac{\partial \mathcal{L}_{\text{photo}}}{\partial \mathbf{I}_{\text{gs}}} \frac{\partial \mathbf{I}_{\text{gs}}}{\partial \mathbf{N}}}_{\text{Gaussian Flow}}, \quad (5)$$

revealing explicit cross-renderer gradient coupling. Since mesh vertices are displaced along $\mathbf{N}$ in Eq. (5), the Gaussian flow directly refines mesh geometry through the shared normal map. The same gradient structure applies to $\mathbf{C}$ and $\mathbf{D}$, enabling coordinated refinement between mesh and 3DGS. The losses $\mathcal{L}_{\text{photo}}$ and $\mathcal{L}_{\text{total}}$ are defined in Eq. (9) and Eq. (8), respectively.

Image Composition. The final image is obtained by compositing the Gaussian and mesh layers via alpha blending [15]:

$$\mathbf{I} = \mathbf{I}_{\text{gs}} + T \cdot \mathbf{I}_{\text{mesh}}, \quad (6)$$

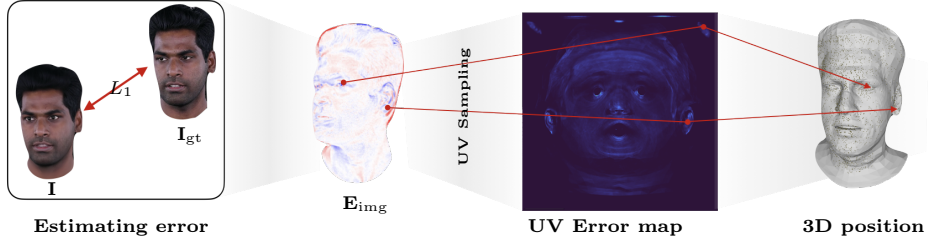


Fig.3: Adaptive Gaussian allocation through error-driven sampling. We project photometric reconstruction errors from image space to UV space, producing an error map (second column) where brighter colors indicate larger errors. Based on this distribution, we sample new Gaussians, shown as 3D point clouds (last column).

where $\mathbf{I}_{\text{gs}}$ in Eq. (4) contributes high-frequency details in the front, and $\mathbf{I}_{\text{mesh}}$ in Eq. (3) provides the base geometry weighted by the transmittance T . This formulation allows Gaussians to refine mesh rendering by adding residual details while preserving the underlying mesh structure.

3.3 Error-Driven Adaptive Gaussian Sampling in UV Space

Our shared UV-parameterized domain provides a surface-aligned space consistent across all renderers. This unified domain naturally enables adaptive Gaussian sampling, as reconstruction errors can be accumulated directly on the UV surface to guide where additional Gaussians are required. Fig. 3 illustrates the resulting sampling distribution and adaptive allocation. Unlike standard 3D Gaussian Splatting [15], which performs gradient-based densification in unstructured 3D space, we sample new Gaussians directly in UV space. Operating in the shared UV domain allows photometric reconstruction errors to be projected back to their corresponding surface locations. We compute the per-pixel photometric error as $\mathbf{E}_{\text{img}} = \|\mathbf{I} - \mathbf{I}_{\text{gt}}\|_1 \in \mathbb{R}^{H \times W}$ and simultaneously render aligned UV coordinates to map these errors onto the UV surface. Regions with high reconstruction error typically indicate missing high-frequency details (*e.g.* wrinkles, pores, subtle facial structures) and thus require additional Gaussian primitives.

Given an input video sequence, we accumulate UV-space reconstruction errors over time and normalize them as

$$w(u, v) = \frac{1}{N} \sum_{t \in T} V(u, v, t) \mathbf{E}_{\text{uv}}(u, v, t), \quad (7)$$

where $V(u, v, t) \in \{0, 1\}$ is a visibility indicator at frame t , equal to 1 if visible and 0 otherwise. $\mathbf{E}_{\text{uv}}(u, v, t)$ denotes the UV-projected photometric error derived from the image-space error $\mathbf{E}_{\text{img}}$, and N is the number of visible observations. Higher values of $w(u, v)$ indicate low reconstruction confidence and thus higher

For notational simplicity, we omit t in other sections. It is also omitted in $p(u, v)$, as the normalized error is accumulated over the entire sequence.

sampling probability. Accordingly, we allocate new Gaussian primitives in proportion to the accumulated error to compensate for the template mesh’s limited capacity to represent fine details. To maintain stable optimization, we cap the number of newly added Gaussians per iteration and prune primitives with low opacity or excessive screen-space extent following [15]. Newly initialized Gaussians sample their attributes from Θ_u and are appended to $\mathcal{G}$.

3.4 Training Objectives

We optimize URHead using a combination of structural, photometric, normal, and geometric regularization losses. The total objective is defined as

$$\mathcal{L}_{\text{total}} = \lambda_{\text{ssim}} \mathcal{L}_{\text{ssim}} + (1 - \lambda_{\text{ssim}}) \mathcal{L}_{\text{photo}} + \lambda_n \mathcal{L}_{\text{normal}} + \lambda_L \mathcal{L}_{\text{lap}}, \quad (8)$$

where $\mathcal{L}_{\text{photo}}$ denotes photometric reconstruction, $\mathcal{L}_{\text{normal}}$ enforces normal consistency, and $\mathcal{L}_{\text{lap}}$ represents Laplacian geometric regularization. The weights λ_n and λ_L balance normal supervision and geometric smoothness, respectively.

The photometric loss consists of ℓ_1 reconstruction and SSIM [35] terms applied to both the composited image and the mesh-only rendering:

$$\mathcal{L}_{\text{photo}} = 0.8 \cdot \|\mathbf{I} - \mathbf{I}_{\text{gt}}\|_1 + 0.2 \cdot \|\mathbf{I}_{\text{mesh}} - \mathbf{I}_{\text{gt}}\|_1, \quad (9)$$

$$\mathcal{L}_{\text{ssim}} = \text{SSIM}(\mathbf{I}, \mathbf{I}_{\text{gt}}), \quad (10)$$

where $\mathbf{I}$ in Eq. (6) denotes the final composited image and $\mathbf{I}_{\text{mesh}}$ in Eq. (3) is the mesh-rendered image. The mesh-only term encourages the template mesh to capture coarse appearance and global texture, while the composited image terms guide high-frequency detail reconstruction and overall visual fidelity. The normal supervision loss is defined as

$$\mathcal{L}_{\text{normal}} = \|\mathbf{N}_{\text{mesh}} - \mathbf{N}_{\text{DAViD}}\|_1, \quad (11)$$

where $\mathbf{N}_{\text{DAViD}}$ is a pseudo ground-truth normal map estimated by the off-the-shelf method DAViD [30]. This term encourages accurate geometric displacement and surface orientation. To suppress noisy displacements, we apply Laplacian mesh regularization:

$$\mathcal{L}_{\text{lap}} = \frac{1}{N} \sum_{i=1}^N \|\delta_i\|^2, \quad (12)$$

where $\delta_i = \mathbf{v}_i - \frac{1}{|\mathcal{A}_i|} \sum_{j \in \mathcal{A}_i} \mathbf{v}_j$ denotes the Laplacian coordinate of vertex $\mathbf{v}_i$, measuring its deviation from the mean of its adjacent vertices $\mathcal{A}_i$.

4 Experimental Results

We conducted extensive experiments on public benchmark datasets, evaluating both quantitative and qualitative results from multiple perspectives. In addition, we analyzed the characteristics of our method through ablation studies. *More results are provided in the supplementary materials.*

4.1 Experimental Setups

Training Details. We used a UV resolution of 2048×2048 for all UV maps. The FLAME model was configured with $N_s = 300$ identity shape coefficients and $N_e = 100$ expression coefficients. For spherical harmonics, we used degree 3 ($K = 16$ coefficients per color channel). The color and normal maps were initialized from VHAP [29] tracking. Gaussians with opacity below 0.01 were pruned during densification, and the occlusion threshold for depth-aware rendering was set to $\epsilon = 0.01$. All models were trained for 50,000 iterations on a single NVIDIA RTX 3090 GPU. The training and rendering times are 120 mins and 62 FPS, respectively.

We used the Adam optimizer [16] and the learning rates are $\{5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 5 \times 10^{-2}, 10^{-5}\}$ for UV maps ($\mathbf{C}, \mathbf{N}, \mathbf{S}, \mathbf{O}, \mathbf{D}$), and $\{10^{-3}, 10^{-5}, 10^{-6}\}$ for FLAME (ψ, θ, t) to ensure training stability. An exponential learning rate decay was applied to the UV parameters after the initial training iterations. Following [44], we evaluated all methods using a head-region mask that matches the FLAME topology for fair comparison.

Datasets. We evaluated our model on three monocular head reconstruction datasets: INSTA [46], PointAvatar [43], and NerFace [8]. All video frames were preprocessed using Robust Video Matting (RVM) [22] for background removal. Each frame was resized to 512×512 resolution. For training and evaluation, we split the frames into two subsets: all frames except the last 350 were used for training, while the final 350 frames were reserved for testing. Our preprocessing followed that of GaussianAvatars [13], which employs VHAP [29] for FLAME parameter tracking, while for other baseline methods, we adopted their respective preprocessing procedures to ensure optimal performance for each approach.

Baselines. We compared our method with six monocular Gaussian-based head avatar methods: SplattingAvatar [31], GaussianAvatars [13], FlashAvatar [38], GaussianBlendShape [26], FateAvatar [41], and RGBAvatar [19]. These methods represent diverse strategies for integrating Gaussians with parametric meshes, including triangle binding [13, 31], UV-space allocation [38], and expression blend-shapes in Gaussian space [19, 26].

4.2 Quantitative Comparisons

Tab. 1 shows the quantitative results on the INSTA [46], PointAvatar [43], and NerFace [8] datasets. Overall, our method consistently outperforms baseline methods across diverse subjects and datasets. On the INSTA dataset [46] with 5 subjects (one subject overlaps with PointAvatar), URHead achieves an average PSNR of 35.1785 (+1.03 over RGBAvatar [19]), SSIM of 0.9675, and LPIPS of 0.0149. Notably, our model achieves the best LPIPS across all subjects compared to the previous best RGBAvatar [19] (0.0251), indicating that our UV-space unification particularly benefits perceptual quality. Our method also improves performance on the PointAvatar and NerFace datasets [8, 43]. The consistent improvements across pixel-level (PSNR), structural (SSIM), and perceptual (LPIPS) metrics confirm that our unified UV-space representation effectively leverages the geometric priors of FLAME [21] while integrating the complementary strengths of both mesh and Gaussian representations. We also

Table 1: Quantitative comparisons. We compare our approach with six state-of-the-art methods. The **best** and **second** results are highlighted. Abbreviations indicate SplattingAvatar (SA) [31], GaussianAvatars (GA) [13], FlashAvatar (FA) [38], GaussianBlendShape (GBS) [26], FateAvatar (FATEA) [41], and RGBAvatar (RGBA) [19].

Datasets		INSTA Dataset [46]						PointAvatar Dataset [43]			NerFace Dataset [8]				
		bala	biden	justin	malte_1	wojtek_1	avg.	yufeng	marcel	avg.	nf_01	nf_02	nf_03	avg.	
PSNR (↑)	SA [31]	32.6698	29.8051	34.9910	27.4599	32.9244	31.5700	26.8354	24.5210	25.6782	28.4617	31.0630	29.8073	29.7773	
	GA [13]	33.9773	31.8956	35.9398	32.0374	33.0825	33.3865	25.3688	26.2412	25.8050	29.5514	33.6670	29.9414	31.0533	
	FA [38]	32.7987	31.1547	35.5084	30.1380	32.4582	32.4116	29.3201	26.5384	27.9293	20.4025	32.7650	20.5508	24.5728	
	GBS [26]	34.7045	31.9907	37.0975	31.6660	34.7700	34.0457	31.0313	24.6815	27.8564	30.4162	32.8143	29.7572	30.9959	
	FATEA [41]	33.6228	30.6082	34.7205	30.4780	34.2138	32.7287	29.9326	24.7649	27.3488	29.5284	32.9680	29.0570	30.5178	
	RGBA [19]	35.1581	31.9064	36.7658	31.6830	35.2092	34.1445	31.4727	25.4436	28.4582	30.6080	34.8521	30.4110	31.9570	
	Ours	35.3185	34.3604	36.5552	34.0563	35.6023	35.1785	26.6888	29.7056	28.1972	30.4711	34.4519	29.9626	31.6285	
SSIM (↑)	SA [31]	0.9325	0.9522	0.9623	0.9294	0.9591	0.9471	0.9052	0.9253	0.9153	0.9306	0.9565	0.9296	0.9389	
	GA [13]	0.9445	0.9681	0.9704	0.9632	0.9632	0.9619	0.9054	0.9373	0.9214	0.9407	0.9606	0.9356	0.9456	
	FA [38]	0.9250	0.9587	0.9654	0.9451	0.9570	0.9502	0.9291	0.9352	0.9322	0.9088	0.9579	0.8954	0.9207	
	GBS [26]	0.9563	0.9693	0.9753	0.9633	0.9728	0.9674	0.9421	0.9359	0.9390	0.9520	0.9573	0.9376	0.9490	
	FATEA [41]	0.9365	0.9617	0.9628	0.9505	0.9663	0.9556	0.9303	0.9346	0.9325	0.9403	0.9595	0.9212	0.9403	
	RGBA [19]	0.9541	0.9718	0.9722	0.9523	0.9663	0.9633	0.9460	0.9379	0.9420	0.9483	0.9678	0.9482	0.9548	
	Ours	0.9585	0.9702	0.9725	0.9649	0.9716	0.9675	0.9135	0.9520	0.9328	0.9490	0.9726	0.9313	0.9510	
LPIPS (↓)	SA [31]	0.0432	0.0415	0.0333	0.0424	0.0261	0.0373	0.0578	0.0844	0.0711	0.0583	0.0211	0.0627	0.0474	
	GA [13]	0.0537	0.0281	0.0239	0.0333	0.0241	0.0326	0.0848	0.0723	0.0785	0.0533	0.0273	0.0562	0.0456	
	FA [38]	0.1044	0.0593	0.0411	0.0681	0.0280	0.0602	0.0471	0.0820	0.0645	0.1120	0.0223	0.1393	0.0912	
	GBS [26]	0.0380	0.0394	0.0247	0.0398	0.0211	0.0326	0.0497	0.0756	0.0626	0.0537	0.0299	0.0592	0.0476	
	FATEA [41]	0.0431	0.0303	0.0239	0.0347	0.0201	0.0304	0.0389	0.0613	0.0501	0.0466	0.0225	0.0549	0.0413	
	RGBA [19]	0.0324	0.0255	0.0193	0.0303	0.0178	0.0251	0.0410	0.0664	0.0537	0.0457	0.0183	0.0469	0.0370	
	Ours	0.0226	0.0123	0.0150	0.0130	0.0117	0.0149	0.0523	0.0392	0.0458	0.0323	0.0168	0.0444	0.0312	

observe lower performance on the *yufeng* subject of PointAvatar [43], whose long hair extends beyond the FLAME topology; we discuss this limitation further in the supplementary material.

4.3 Qualitative Comparisons

Fig. 4 shows the qualitative comparison with baselines. In the *bala* sequence, our method faithfully reconstructs teeth, while baselines either blur or hallucinate the oral region. For *biden*, RGBA [19] and FATEA [41] over-exaggerate facial features, while GBS [26] under-represents them due to the limited expressiveness of the predefined blendshape space. In *wojtek_1*, our method is the only one that captures facial moles, and the *malte_1* sequence shows that ours reproduces subtle wrinkles more accurately. In *nf_01*, baseline methods produce noticeably blurred renderings, whereas ours preserves vivid appearance. Finally, in *justin*, our method reconstructs forehead wrinkles with temporal consistency, while RGBA [19] partially captures forehead wrinkles but lacks overall consistency in wrinkle rendering across frames. These results suggest that our unified UV-space formulation effectively captures fine-grained details through adaptive Gaussian sampling while maintaining structural coherence via the mesh component. Full comparisons across all subjects are provided in the supplementary material. Fig. 5 shows that the mesh can capture base geometry, but the results lack fine details (*e.g.* teeth, skin texture), while our model recovers them via Gaussian residuals, underscoring the value of our hybrid design.

Cross-Subject Reenactment. Fig. 6 evaluates the generalization capability of our method through cross-subject reenactment experiments. Our method successfully transfers the facial expressions of target while preserving the identity-specific features of source. The unified UV-space representation enables smooth expression transfer without artifacts.

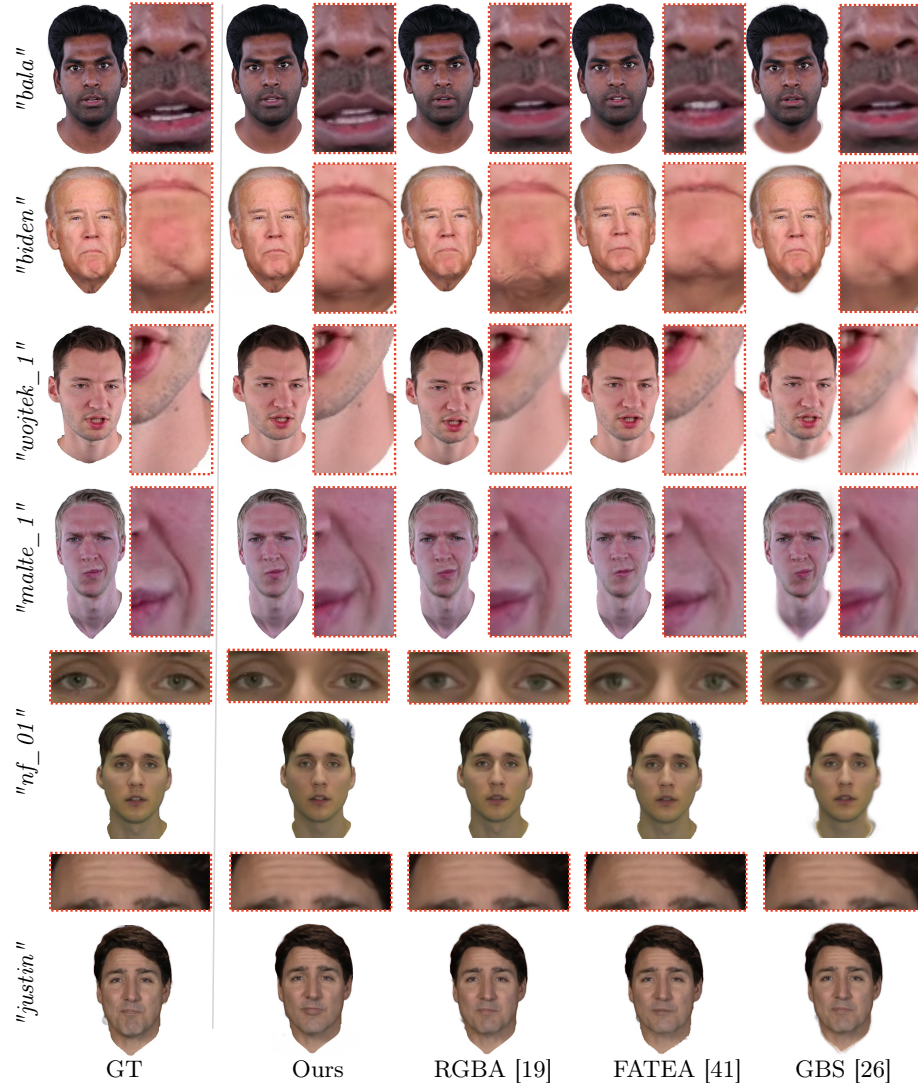


Fig. 4: Qualitative comparisons. Our method produces photorealistic renderings with accurate fine details (teeth, wrinkles, skin dot, eyes) compared to four baselines. Zoom-in views highlight our detail preservation.

Table 2: Ablation study of our model on the INSTA dataset [46].

Method	PSNR	SSIM	LPIPS
w/o 3DGS	33.60	0.943	0.057
w/o $\mathcal{L}_{\text{normal}}$	34.45	0.948	0.047
w/o occlusion-aware blending	33.63	0.949	0.045
w/o $0.2 \cdot \ \mathbf{I}_{\text{mesh}} - \mathbf{I}_{\text{gt}}\ _1$	34.90	0.955	0.031
Ours	35.31	0.958	0.022

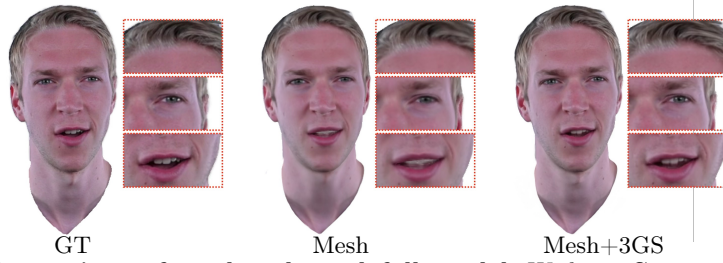


Fig. 5: Comparison of mesh only and full model. Without Gaussians (Mesh only), the rendering lacks fine details such as teeth visibility and skin texture. Our full model (Mesh+3GS) captures these high-frequency details through Gaussian residuals.



Fig. 6: Cross-subject reenactment. The visualization shows avatars trained on one identity (sources) are driven by facial expressions from different subjects (target). Multiple source identities are driven by expressions from a single target subject.

4.4 Ablation Studies

Component Analysis. Tab. 2 analyzes the contribution of each component in our framework. At the first row, the model without 3DGS causes the most significant degradation in perceptual quality, confirming that the fixed-topology mesh alone cannot capture high-frequency appearance details. At the second row, without $\mathcal{L}_{\text{normal}}$, geometric accuracy degrades as the displacement map loses guidance from the pseudo ground-truth normals. At the third row, disabling occlusion-aware blending allows Gaussians behind the mesh surface to be rendered redundantly, introducing visual inconsistencies in the composited image. At the fourth row, removing the $0.2 \cdot \|\mathbf{I}_{\text{mesh}} - \mathbf{I}_{\text{gt}}\|_1$ weakens the mesh’s incentive to learn meaningful texture and geometry on its own, forcing the Gaussians to compensate for both base appearance and residual details, which reduces overall perceptual quality. We note that the model without 3DGS can occasionally surpass other models on certain subjects, as Gaussians may introduce geometric disorder; we analyze this further below.

Uniform vs. Adaptive Sampling. Tab. 3 compares uniform and adaptive sampling strategies across UV resolutions. With uniform sampling, performance peaks at 512×512 , but degrades at 1024×1024 , indicating overfitting from redundant Gaussians. In contrast, adaptive sampling shows improvement with increasing resolution, validating that our adaptive gaussian sampling efficiently concentrates Gaussians where most needed.

Stability of Animation-Driven Deformation. To verify animation stability, we report frame-to-frame LPIPS, defined as $\frac{1}{N-1} \sum_{t=1}^{N-1} \text{LPIPS}(\hat{I}_t, \hat{I}_{t+1})$, where $\hat{I}_t$ is the predicted frame at time t , on the INSTA dataset. As shown in Tab. 4, our method achieves the lowest frame-to-frame LPIPS mean (0.0153) and vari-

Table 3: Design validation of sampling strategies by changing UV resolutions.

Uniform	PSNR	SSIM	LPIPS	Adaptive	PSNR	SSIM	LPIPS
128 ²	33.49	0.9447	0.047	256 ²	34.00	0.948	0.042
256 ²	33.94	0.9482	0.038	512 ²	34.15	0.949	0.040
512 ²	34.11	0.949	0.034	1024 ²	34.17	0.949	0.0412
1024 ²	33.49	0.948	0.0370	2048 ²	34.37	0.950	0.031

Table 4: Frame-to-frame LPIPS consistency on the INSTA dataset.

	FATEA	GBS	RGBA	Ours
LPIPS Mean ↓	0.0325	0.0312	0.0267	0.0153
LPIPS Var ↓	0.000026	0.000033	0.000032	0.000010

ance (0.000010), indicating that our unified UV-space representation produces temporally consistent renderings under animation-driven deformation. The low variance further confirms that our method maintains stable rendering quality across frames without introducing flickering or temporal artifacts.

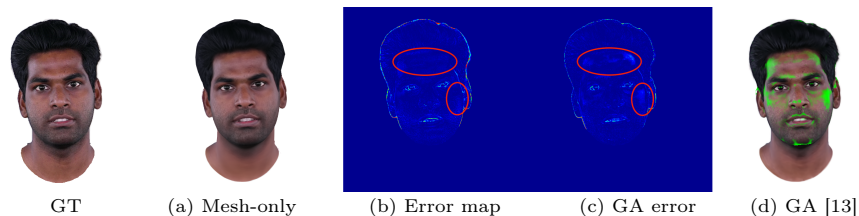


Fig. 7: Geometric disorder in Gaussian-based baselines. (a) Mesh-only rendering from our method. (b) Corresponding error map. (c) Error map of a GA, representative of a common failure in mesh-free Gaussian methods. (d) Rendering from GA with a green mesh embedded inside the head to expose opacity collapse. Visible green regions indicate Gaussians failing to cover the facial surface.

Geometric Disorder. Fig. 7 visualizes a failure common to Gaussian-based methods, exemplified by GA. To visualize this phenomenon, we embed a green mesh inside the head; when Gaussians converge to low opacity, the underlying mesh becomes visible through the face as green regions shown in (d). This occurs because Gaussians fail to place properly on the facial surface and instead scatter behind the head. In contrast, our mesh component provides a reliable geometric prior that prevents such disorder.

5 Conclusion

We have proposed URHead, a unified approach that combines mesh geometry and Gaussian appearance modeling in UV space for photorealistic 3D head avatar reconstruction. This is the first attempt to jointly optimize the mesh and Gaussian representations in the shared UV-parameterized space. This shared representation leverages mesh stability for base geometry while Gaussians capture details with adaptive sampling. Extensive experiments demonstrate competitive performance, validating the effectiveness of our unified approach. We believe that these shared representations between different renderers open new directions for combining classical and neural rendering in avatar generation to take advantage of different renderers.

Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2021-II211341).

References

1. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 157–164 (2023)
2. Cai, H., Xiao, Y., Wang, X., Li, J., Guo, Y., Fan, Y., Gao, S., Zhang, J.: Hera: Hybrid explicit representation for ultra-realistic head avatars. In: *CVPR*. pp. 260–270 (2025)
3. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* **20**(3), 413–425 (2013)
4. Chen, Y., Wang, L., Li, Q., Xiao, H., Zhang, S., Yao, H., Liu, Y.: Monogaussianavatar: Monocular gaussian point-based head avatar. In: *ACM SIGGRAPH*. pp. 1–9 (2024)
5. Cheng, K.H., Tsai, C.C.: Children and parents’ reading of an augmented reality picture book: Analyses of behavioral patterns and cognitive attainment. *Computers & Education* **72**, 302–312 (2014)
6. Debevec, P.: The light stages and their applications to photoreal digital actors (2012)
7. Feng, Y., Feng, H., Black, M.J., Bolkart, T.: Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics* **40**(4), 1–13 (2021)
8. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: *CVPR*. pp. 8649–8658 (2021)
9. Giebenhain, S., Kirschstein, T., Georgopoulos, M., Rünz, M., Agapito, L., Nießner, M.: Learning neural parametric head models. In: *CVPR*. pp. 21003–21012 (2023)
10. Grassal, P.W., Prinzler, M., Leistner, T., Rother, C., Nießner, M., Thies, J.: Neural head avatars from monocular rgb videos. In: *CVPR*. pp. 18653–18664 (2022)
11. Guo, K., Lincoln, P., Davidson, P., Busch, J., Yu, X., Whalen, M., Harvey, G., Orts-Escolano, S., Pandey, R., Dourgarian, J., et al.: The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics* **38**(6), 1–19 (2019)
12. Healey, J., Wang, D., Wigington, C., Sun, T., Peng, H.: A mixed-reality system to promote child engagement in remote intergenerational storytelling. In: *ISMAR-Adjunct*. pp. 274–279 (2021)
13. Hu, L., Zhang, H., Zhang, Y., Zhou, B., Liu, B., Zhang, S., Nie, L.: Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In: *CVPR*. pp. 634–644 (2024)
14. Kachach, R., Perez, P., Villegas, A., Gonzalez-Sosa, E.: Virtual tour: An immersive low cost telepresence system. In: *VR Workshop*. pp. 504–506 (2020)
15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42**(4), 139–1 (2023)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)

17. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics* **39**(6) (2020)
18. Lee, J., Ahn, J., Kang, T., Choo, J.: Texavatars: Hybrid texel-3d representations for stable rigging of photorealistic gaussian head avatars. *arXiv preprint arXiv:2512.21099* (2025)
19. Li, L., Li, Y., Weng, Y., Zheng, Y., Zhou, K.: Rgbavatar: Reduced gaussian blendshapes for online modeling of head avatars. In: *CVPR*. pp. 10747–10757 (2025)
20. Li, N., Zhang, Z., Liu, C., Yang, Z., Fu, Y., Tian, F., Han, T., Fan, M.: Vmirror: Enhancing the interaction with occluded or distant objects in vr with virtual mirrors. In: *CHI*. pp. 1–11 (2021)
21. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics* **36**(6), 194–1 (2017)
22. Lin, S., Yang, L., Saleemi, I., Sengupta, S.: Robust high-resolution video matting with temporal guidance (2021)
23. Lombardi, S., Saragih, J., Simon, T., Sheikh, Y.: Deep appearance models for face rendering. *ACM Transactions on Graphics* **37**(4), 1–13 (2018)
24. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: *3DV*. pp. 800–809 (2024)
25. Ma, S., Luo, Y., Yang, W., Yang, Y.: Mags: Reconstructing and simulating dynamic 3d objects with mesh-adsorbed gaussian splatting. In: *ICCV*. pp. 8745–8755 (2025)
26. Ma, S., Weng, Y., Shao, T., Zhou, K.: 3d gaussian blendshapes for head avatar animation. In: *ACM SIGGRAPH*. pp. 1–10 (2024)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)
28. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3d face model for pose and illumination invariant face recognition. In: *AVSS*. pp. 296–301 (2009)
29. Qian, S.: Vhap: Versatile head alignment with adaptive appearance priors (sep 2024). <https://doi.org/10.5281/zenodo.14988309>, <https://github.com/ShenhanQian/VHAP>
30. Saleh, F., Aliakbarian, S., Hewitt, C., Petikam, L., Xiao-Xian, Criminisi, A., Cashman, T.J., Baltrušaitis, T.: DAViD: Data-efficient and accurate vision models from synthetic data (2025)
31. Shao, Z., Wang, Z., Li, Z., Wang, D., Lin, X., Zhang, Y., Fan, M., Wang, Z.: Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In: *CVPR*. pp. 1606–1616 (2024)
32. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. *Acm Transactions on Graphics* **38**(4), 1–12 (2019)
33. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: *CVPR*. pp. 2387–2395 (2016)
34. Tran, L., Liu, X.: Nonlinear 3d face morphable model. In: *CVPR*. pp. 7346–7355 (2018)
35. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
36. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: *CVPR*. pp. 20310–20320 (2024)

37. Xiang, F., Xu, Z., Hasan, M., Hold-Geoffroy, Y., Sunkavalli, K., Su, H.: Neutex: Neural texture mapping for volumetric neural rendering. In: CVPR. pp. 7119–7128 (2021)
38. Xiang, J., Gao, X., Guo, Y., Zhang, J.: Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In: CVPR. pp. 1802–1812 (2024)
39. Yang, H., Zheng, M., Feng, W., Huang, H., Lai, Y.K., Wan, P., Wang, Z., Ma, C.: Towards practical capture of high-fidelity relightable avatars. In: SIGGRAPH Asia. pp. 1–11 (2023)
40. Zackariasson, P., Wilson, T.L.: The video game industry: Formation, present state, and future. Routledge (2012)
41. 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. pp. 5535–5545 (2025)
42. 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. pp. 13545–13555 (2022)
43. Zheng, Y., Yifan, W., Wetzstein, G., Black, M.J., Hilliges, O.: Pointavatar: Deformable point-based head avatars from videos. In: CVPR. pp. 21057–21067 (2023)
44. Zielonka, W., Bolkart, T., Beeler, T., Thies, J.: Gaussian eigen models for human heads. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15930–15940 (2025)
45. Zielonka, W., Bolkart, T., Thies, J.: Towards metrical reconstruction of human faces. In: ECCV. pp. 250–269 (2022)
46. Zielonka, W., Bolkart, T., Thies, J.: Instant volumetric head avatars. In: CVPR. pp. 4574–4584 (2023)