GDAVATAR: GEOMETRY-GUIDED DYNAMIC GAUSSIAN HEAD AVATARS FROM MONOCULAR VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent Gaussian head avatars achieve remarkable rendering quality and real-time performance, but coarse canonical geometry and limited motion information hinder faithful reconstruction of expression-dependent appearance details. In this paper, we propose GDAvatar, a geometry-guided dynamic Gaussian head avatar framework for monocular videos. We first optimize the canonical mesh using a vertex-bound Gaussian representation to obtain personalized geometry. Building upon the optimized geometry, we then employ a triangle-bound Gaussian representation to model dynamic Gaussian attribute offsets by jointly encoding canonical and dynamic features. An attention fusion module adaptively balances these features, enabling more accurate modeling of local non-rigid deformations. Furthermore, we construct a continuous editable Gaussian map from the learned representation, enabling flexible and high-quality texture editing. Extensive experiments on multiple datasets demonstrate that GDAvatar achieves superior rendering quality and animation fidelity compared with state-of-the-art methods, while naturally supporting high-quality texture editing.

1 INTRODUCTION

High-fidelity digital head avatars have attracted increasing attention in virtual reality Weidner et al. (2023), digital humans, and film production. Traditional approaches Yang et al. (2023) often rely on multi-view capture systems or specialized scanners, whose high cost and complex setup limit practical use. In contrast, monocular video provides a simple and accessible capture setup (Zheng et al., 2022; Zielonka et al., 2023). With its explicit representation and efficient rasterization, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has recently become a promising representation for monocular head avatar reconstruction.

Most 3DGS-based head avatar methods (Shi et al., 2026; Kabadayi et al., 2026) adopt parametric face models such as FLAME (Li et al., 2017) as geometric and motion priors. By binding Gaussians to a template mesh and applying linear blend skinning (LBS), these methods inherit the motion prior while reducing the ambiguity of monocular reconstruction. However, the canonical template cannot fully capture identity-specific geometry, such as hair, neck shape, and individual facial structures. Existing methods (Dhamo et al., 2024; Shao et al., 2024; Xiang et al., 2023) mainly compensate for these discrepancies through Gaussian position optimization, while GaussianAvatars (Qian et al., 2024) estimates coarse static vertex offsets and MeGA (Wang et al., 2025a) introduces additional unconstrained Gaussians for regions such as hair. STAvatar (Zhao et al., 2026) further models spatially varying Gaussian attributes with a UV Offset Branch, but relies on externally preprocessed personalized geometry. Although these approaches increase representation flexibility, allowing multiple Gaussian attributes to compensate for the same reconstruction error can entangle persistent identity-specific geometry with expression-dependent variations.

We identify this representation ambiguity between identity-specific geometry and expression-dependent variations in monocular Gaussian head avatar reconstruction. To address it, we propose GDAvatar, a geometry-guided framework that progressively allocates Gaussian degrees of freedom from canonical geometry to dynamic variations. We first optimize personalized canonical geometry using vertex-bound Gaussians with constrained degrees of freedom, preventing appearance-related attributes from compensating for geometric discrepancies. After fixing the learned geometry, we release additional degrees of freedom with triangle-bound Gaussians and model expression-dependent

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

variations through a geometry-conditioned UV Offset Branch. Finally, we construct a continuous UV Gaussian representation that preserves the learned dynamic variations for appearance editing.

Our contributions are summarized as follows:

- We identify the representation ambiguity between identity-specific geometry and expression-dependent variations in monocular Gaussian head avatar reconstruction, and resolve it by progressively allocating Gaussian degrees of freedom from personalized geometry to dynamic appearance.
- We introduce a geometry-conditioned dynamic Gaussian representation that uses the learned personalized canonical geometry as an identity-specific basis and adaptively integrates canonical and dynamic features to predict expression-dependent Gaussian attribute offsets.
- We develop a dynamic-preserving continuous UV Gaussian representation that retains expression-dependent Gaussian variations in a continuous UV space, enabling appearance editing without discarding the learned dynamic representation.

2 Related Work

2.1 3D Head Avatar Reconstruction

With the development of neural rendering, head avatar reconstruction has evolved from mesh-based methods Grassal et al. (2022); Bharadwaj et al. (2023); Wang et al. (2023b) to NeRF-based Zheng et al. (2022); Zielonka et al. (2023), point-based Wang et al. (2023a); Zheng et al. (2023), and more recently Gaussian-based methods Qian et al. (2024); Shao et al. (2024). NeRF Mildenhall et al. (2021) has advanced high-fidelity head avatar reconstruction. IMAvatar Zheng et al. (2022) learns animatable avatars with implicit geometry and texture, while INSTA Zielonka et al. (2023) combines dynamic radiance fields with parametric face models for efficient training. PointAvatar Zheng et al. (2023) uses deformable points and learned deformation fields for controllable animation. However, NeRF methods rely on costly volumetric rendering, while point-based methods are limited by discrete representations, making it difficult to jointly achieve high quality, efficient training, and real-time performance.

The emergence of 3DGS Kerbl et al. (2023) has inspired a series of Gaussian-based approaches Dhamo et al. (2024); Peng et al. (2025b); Wang et al. (2025c;b); Feng et al. (2025) for head avatar reconstruction. GaussianAvatars Qian et al. (2024) and SplattingAvatar Shao et al. (2024) bind Gaussians to the FLAME mesh, allowing animation through mesh deformation. Several methods Xiang et al. (2023); Peng et al. (2025a) further introduce dynamic offsets for non-rigid deformation. MonoAvatar Chen et al. (2024) and GaussianHeadAvatar Xu et al. (2024) predict Gaussian attributes with neural networks, while GaussianBlendshapes Ma et al. (2024) and RGBAvatar Li et al. (2025) construct Gaussian blendshapes for flexible expression modeling. Due to the discrete nature of Gaussian representations, recent works explore continuous UV-space representations. MeGA Wang et al. (2025a) predicts a UV displacement map to refine mesh geometry, while Gaussian Deja-vu Yan et al. (2024), FateAvatar Zhang et al. (2025) and GTAvatar Baert et al. (2026) adopt UV-space representations for Gaussian attribute modeling. STAvatar Zhao et al. (2026) further learns per-Gaussian attribute offsets in UV space for fine-grained detail modeling.

2.2 Geometry Priors in Head Avatars

Early implicit-representation methods Zheng et al. (2022); Zielonka et al. (2023) lack explicit geometric structures, limiting controllable animation. To incorporate geometric guidance, HHAvatar Liao et al. (2025) adopts meshes as geometric priors but relies on neural deformation without explicit parametric motion priors. In contrast, 3D Morphable Models (3DMMs) Li et al. (2017); Gerig et al. (2018); Palafox et al. (2021), such as FLAME Li et al. (2017), provide explicit correspondence and controllable deformation priors, and are widely adopted for head avatar reconstruction Tang et al. (2024); Liu et al. (2025); Saunders et al. (2025); Zhang et al. (2025). However, FLAME cannot fully capture subject-specific geometry. To address this limitation, several works have explored refining the FLAME template. GaussianAvatars Qian et al. (2024) adopts VHAP Qian

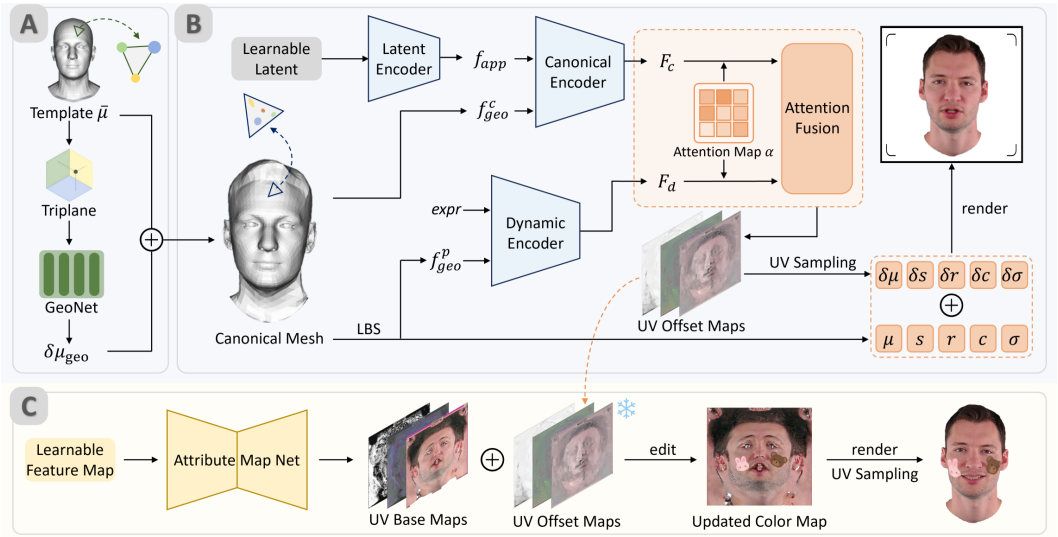

Figure 1: Overview of GDAvatar. Our framework consists of three stages. (A) Canonical Geometry Optimization employs vertex-bound Gaussians, and GeoNet predicts vertex offsets to personalize the canonical mesh. (B) Dynamic Appearance Modeling employs triangle-bound Gaussians, where a UV Offset Branch predicts Gaussian attribute offsets by integrating canonical priors and dynamic features. The two stages A and B are optimized progressively for high-fidelity avatar reconstruction. (C) Editable Gaussian Map Construction builds a continuous UV attribute map to support flexible appearance editing.

(2024) preprocessing to optimize the FLAME template with static vertex offsets, and the refined geometry has been widely adopted by subsequent methods Sun et al. (2025); Wang et al. (2025c); Zhao et al. (2026). GeoAvatar Moon et al. (2025) introduces explicit inner-mouth geometry, while MeGA Wang et al. (2025a) predicts UV-space mesh displacements for facial geometry refinement and uses free Gaussians for hair. However, existing approaches mainly rely on additional geometric structures or subtle mesh adjustments to compensate for FLAME's limitations, which still lack flexibility in representing diverse subject-specific geometry.

3 Method

GDAvatar adopts a progressive Gaussian degree-of-freedom allocation strategy to separate identity-specific geometry from expression-dependent appearance. The key idea is to first restrict Gaussian degrees of freedom to establish a personalized canonical representation, and then progressively release additional degrees of freedom for modeling dynamic variations. As illustrated in Fig. 1, Stage A establishes personalized canonical geometry using vertex-bound Gaussians with constrained degrees of freedom and topology-preserving regularization. Stage B then releases additional degrees of freedom through triangle-bound Gaussians and introduces a geometry-conditioned UV Offset Branch that adaptively fuses canonical and dynamic features to model expression-dependent attribute variations. Finally, Stage C transfers the learned Gaussian attributes to a continuous editable UV representation while preserving the learned dynamic variations.

3.1 Canonical Geometry Optimization

Most FLAME-based head avatar methods Chen et al. (2024); Li et al. (2025) use FLAME as the canonical geometric prior, but its fixed template cannot fully capture identity-specific structures such as hair, neck shape, and fine facial details. Directly optimizing fully parameterized face-bound Gaussians can introduce representation ambiguity, as appearance attributes may compensate for geometric errors. We therefore first optimize personalized canonical geometry with vertex-bound Gaussians under restricted degrees of freedom before modeling dynamic variations.

Vertex-bound Gaussian Representation. To isolate geometric optimization from appearance modeling, we establish a one-to-one correspondence between mesh vertices and Gaussians:

$$G_{geo} = \{\mu_g, r_g, s_g, c_g, \sigma_g\}, \tag{1}$$

where μ_g, r_g, s_g, c_g, and σ_g denote the Gaussian center, rotation, scale, color, and opacity, respectively. During canonical geometry optimization, we fix $r_g = I$ and $\sigma_g = 1$, use a learnable isotropic scale s_g, and represent c_g with 0-degree spherical harmonics. Thus, the Gaussian center serves as the primary geometric degree of freedom, while the restricted appearance parameters have limited capacity to compensate for geometric discrepancies.

Canonical Mesh Refinement. To improve geometric resolution while maintaining efficient optimization, we subdivide the original 5K-vertex FLAME mesh once, yielding approximately 20K vertices $\bar{\mu}$. The subdivided template provides the fixed topology and initializes the Gaussian centers. For each vertex, a learnable triplane mapping Φ_{tri} extracts spatial features, which are processed by GeoNet Φ_{geo} to predict a geometric offset:

$$\Delta\mu_g = \Phi_{\text{geo}}\left(\Phi_{\text{tri}}(\bar{\mu})\right). \tag{2}$$

The personalized canonical mesh is then obtained as

$$\mu_c = \bar{\mu} + \Delta\mu_g. \tag{3}$$

We optimize the geometric offsets with reconstruction and topology-preserving regularization losses. After Stage A, the optimized canonical mesh μ_c is fixed, and Stage B introduces additional Gaussian degrees of freedom to model expression-dependent appearance. This sequential optimization assigns identity-specific variations to the canonical geometry before introducing dynamic variations, providing a personalized basis for subsequent dynamic modeling.

3.2 Dynamic Appearance Modeling

After establishing the personalized canonical geometry, we progressively release the remaining Gaussian degrees of freedom through triangle-bound Gaussians for high-fidelity dynamic appearance modeling. The canonical geometry remains fixed throughout this stage, while the triangle-bound representation provides full-DoF local attribute modeling and adaptive densification. Based on this personalized geometry, we further introduce a geometry-conditioned UV Offset Branch to model expression-dependent Gaussian attribute variations.

Triangle-bound Gaussian Representation. We uniformly sample points in the UV space. For a sampled point with barycentric coordinates $(\lambda_1, \lambda_2, \lambda_3)$ satisfying $\lambda_1 + \lambda_2 + \lambda_3 = 1$, its 3D position is computed as

$$\mu = \lambda_1 v_1 + \lambda_2 v_2 + \lambda_3 v_3, \tag{4}$$

where v_1, v_2, and v_3 are the world-space coordinates of the corresponding triangle vertices. For each triangle, we construct a local coordinate system with rotation matrix R and scale factor k. The local Gaussian rotation r_l and anisotropic scale s_l are transformed into the world coordinate system as

$$r = Rr_l, \qquad s = ks_l. \tag{5}$$

The color c and opacity σ are also learnable, forming the base Gaussian set

$$G_{\text{base}} = \{\mu, r, s, c, \sigma\}. \tag{6}$$

Adaptive densification further clones and splits Gaussians to improve local detail representation, with each newly generated Gaussian retaining a UV coordinate for subsequent attribute modeling.

Although triangle-bound Gaussians provide flexible local attributes, mesh parameterization alone cannot adequately capture expression-dependent non-rigid variations, particularly around the lips and eyelids. We therefore introduce a UV Offset Branch to predict spatially varying Gaussian attribute offsets. Unlike STAvatar Zhao et al. (2026), which relies on externally preprocessed personalized geometry, our branch operates on the canonical geometry learned within the reconstruction framework and incorporates a learnable latent representation with adaptive fusion of canonical and dynamic features.

Canonical and Dynamic Feature Learning. The UV Offset Branch extracts canonical identity features and dynamic expression features from separate inputs. For canonical feature learning, we

Method	INSTA			PointAvatar			NerFace		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
FlashAvatar (CVPR'2024)	27.368	0.933	0.060	26.443	0.909	0.092	31.335	0.961	0.035
MonoAvatar (SIGGRAPH'2024)	27.586	0.935	0.091	28.075	0.932	0.098	33.643	0.973	0.031
GaussianAvatars (CVPR'2024)	26.817	0.942	0.079	24.321	0.886	0.114	28.725	0.954	0.040
RGBAvatar (CVPR'2025)	30.640	0.956	0.047	25.268	0.892	0.108	30.339	0.959	0.034
FateAvatar (CVPR'2025)	27.804	0.943	0.058	28.596	0.933	0.073	33.503	0.972	0.027
STAvatar (CVPR'2026)	30.360	0.957	0.045	26.103	0.914	0.091	30.120	0.962	0.038
Ours	**31.431**	**0.960**	**0.043**	**29.184**	**0.936**	**0.063**	**34.162**	**0.975**	**0.021**

Table 1: Quantitative comparison on the INSTA, PointAvatar, and NerFace datasets. Best results are highlighted in bold.

introduce a learnable latent map $Z \in \mathbb{R}^{3\times H\times W}$. A learnable U-Net $\Phi_l(\cdot)$ extracts the appearance feature f_{app}, while the personalized canonical mesh μ_c is projected onto the UV space to generate a canonical position map $P_c \in \mathbb{R}^{3\times H\times W}$. Its geometric feature f^c_{geo} is then combined with f_{app} to obtain the Canonical Feature:

$$f_{app} = \Phi_l(Z), \qquad F_c = \Phi_c([f_{app}, f^c_{geo}]). \tag{7}$$

For dynamic feature learning, the posed mesh is projected onto the UV space to generate a position map $P_p \in \mathbb{R}^{3\times H\times W}$, from which the posed geometric feature f^p_{geo} is extracted. The expression parameters e are jointly encoded with f^p_{geo} by a deformable feature encoder $\Phi_d(\cdot)$ based on Deformable ConvNets v2 Zhu et al. (2019):

$$F_d = \Phi_d(f^p_{geo}, e; \Delta p, \Delta m), \tag{8}$$

where Δp and Δm denote the learned sampling offsets and modulation scalars. Expression, pose, and camera parameters are jointly refined during training.

UV Offset Map Based on Attention Fusion. The released dynamic degrees of freedom are modeled through a geometry-conditioned UV Offset Branch. The Canonical Feature F_c and Dynamic Feature F_d encode complementary identity and expression information, whose relative importance varies across facial regions. We therefore introduce an Attention Fusion module to adaptively combine them:

$$\begin{aligned} \alpha &= \Phi_{gate}([F_c, F_d]), \\ F_f &= \alpha \odot F_d + (1-\alpha) \odot F_c, \end{aligned} \tag{9}$$

where $\Phi_{gate}(\cdot)$ denotes a lightweight gating network and $\alpha \in [0,1]^{H\times W}$ is the attention map. The fused feature is decoded into the UV offset map $O_{off} \in \mathbb{R}^{13\times H\times W}$:

$$O_{off} = \Phi_{fuse}(F_f). \tag{10}$$

Bilinear sampling at each Gaussian's UV coordinate produces the attribute offsets

$$\Delta G = \{\Delta\mu, \Delta s, \Delta r, \Delta c, \Delta\sigma\}, \tag{11}$$

and the final Gaussian attributes are computed as

$$G_{app} = G_{base} + \Delta G. \tag{12}$$

With the personalized canonical geometry fixed, the released degrees of freedom are used to model expression-dependent variations rather than re-learning identity-specific geometry.

3.3 Editable Gaussian Map Construction

The reconstruction representation combines discrete Gaussian primitives with spatially varying, expression-dependent attribute offsets. While effective for rendering, it does not provide a continuous representation for direct appearance editing. We therefore construct a continuous UV Gaussian representation that preserves the learned dynamic offsets while converting the base Gaussian attributes into an editable UV domain.

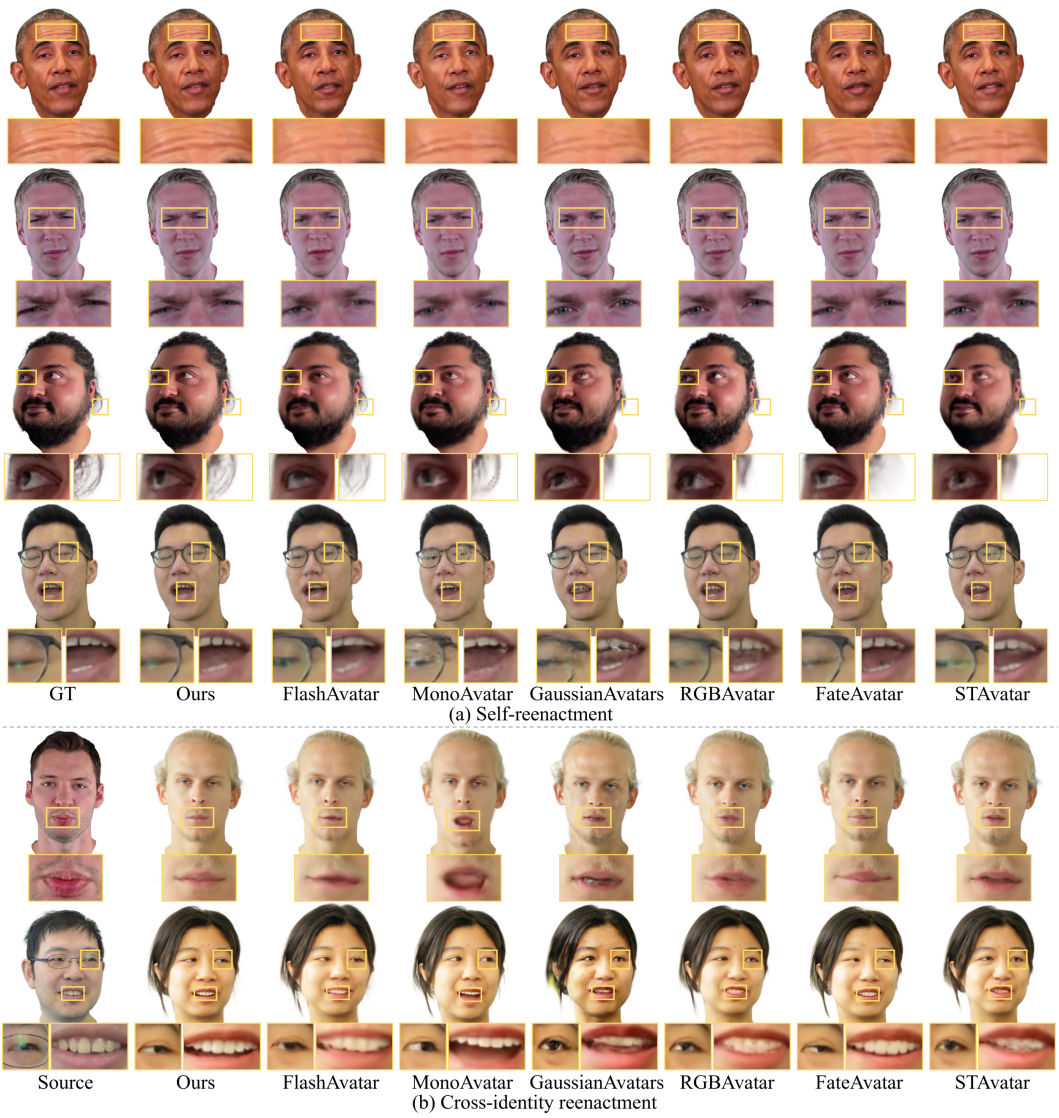

Figure 2: Qualitative comparison on the INSTA, PointAvatar, and NerFace datasets. (a) Self-reenactment. (b) Cross-identity reenactment. Our method preserves fine details and achieves accurate pose transfer across identities.

We freeze the learned UV Offset Branch and introduce an Attribute Map Net Φ_{map} to reconstruct the continuous base Gaussian attributes. Given a learnable feature map $F \in \mathbb{R}^{10\times H\times W}$, the base attribute map is obtained as

$$O_{base}^{attr} = \Phi_{map}(F), \tag{13}$$

where $O_{base}^{attr} \in \mathbb{R}^{10\times H\times W}$. Since the base Gaussian positions are defined in the UV space, we concatenate the predicted attributes with the UV base position map to form the complete base map $O_{base} \in \mathbb{R}^{13\times H\times W}$. The learned dynamic offset map is retained and combined with the base map:

$$O_{edit} = O_{base} + O_{off}. \tag{14}$$

For both original and densified Gaussians, their UV coordinates are used to bilinearly sample the corresponding attributes from O_{edit}, providing a unified continuous representation. This construction extends the dynamic Gaussian representation from Stage B to a continuous UV domain, enabling appearance editing without discarding the learned expression-dependent variations.

Appearance editing can be performed directly on the color component O^c_{edit}. For example, local and global transformations are formulated as

$$O^c_{edit}{}' = \begin{cases} M \odot C + (1 - M) \odot O^c_{edit}, & \text{local,} \\ \Phi_{style}(O^c_{edit}), & \text{global,} \end{cases} \tag{15}$$

where M denotes the editing mask, C denotes the target texture, and Φ_{style} represents a pretrained style transfer model (Johnson et al., 2016). The edited UV map is sampled back to Gaussian attributes for rendering, providing a continuous editing interface while preserving the learned dynamic representation.

3.4 Optimization

We optimize the three stages with objectives tailored to their respective roles. Stage A focuses on personalized canonical geometry, Stage B on expression-dependent Gaussian appearance, and Stage C on continuous UV attribute reconstruction.

Geometry Optimization. In Stage A, we optimize the canonical geometry using the photometric reconstruction losses $\mathcal{L}_1$ and $\mathcal{L}_{dssim}$, together with parameter and Laplacian regularization terms.

To constrain vertex deformation and Gaussian scale during canonical geometry optimization, we define

$$\mathcal{L}_{reg} = \|\Delta\mu_g\|_2^2 + \|s_g\|_2^2 . \tag{16}$$

To further preserve the local geometric structure of the FLAME template and encourage smooth scale variation, we introduce

$$\mathcal{L}_{lap} = \|L(\mu_c) - L(\bar{\mu})\|_2^2 + \|L(s_g)\|_2^2 , \tag{17}$$

where $L(\cdot)$ denotes the Laplacian operator. The first term constrains local mesh deformation relative to the initial template, while the second promotes spatially smooth Gaussian scales. The overall geometry optimization objective is

$$\mathcal{L}_{geo} = \lambda_1\mathcal{L}_1 + \lambda_{dssim}\mathcal{L}_{dssim} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{lap}\mathcal{L}_{lap}, \tag{18}$$

where $\lambda_1 = 0.8$, $\lambda_{dssim} = 0.2$, $\lambda_{reg} = 10$, and $\lambda_{lap} = 10^5$.

Appearance Optimization. In Stage B, we optimize the Gaussian attributes using photometric reconstruction and perceptual losses, together with scale and attribute-offset regularization:

$$\mathcal{L}_{app} = \lambda_2\mathcal{L}_1 + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_s\mathcal{L}_s + \lambda_{off}\mathcal{L}_{off}. \tag{19}$$

To prevent degenerate anisotropic Gaussians, we constrain the ratio between the maximum and minimum scale components of each base Gaussian:

$$\mathcal{L}_s = \frac{1}{N}\sum_{i=1}^{N} \max\left(\frac{s_i^{\max}}{s_i^{\min}} - \tau,\, 0\right), \tag{20}$$

where $\tau = 9$. We further regularize the predicted scale and color offsets to avoid excessive deviations from the base Gaussian attributes:

$$\mathcal{L}_{off} = \|\Delta s\|_1 + \|\Delta c\|_1 . \tag{21}$$

The loss weights are set to $\lambda_2 = 1$, $\lambda_{vgg} = 0.1$, $\lambda_s = 0.1$, and $\lambda_{off} = 10^{-3}$.

In Stage C, we freeze the learned UV Offset Branch and reconstruct the continuous UV attribute map using the first three terms of $\mathcal{L}_{app}$. This stage distills the learned base Gaussian attributes into the continuous UV representation while retaining the expression-dependent offset map learned in Stage B.

4 Experiments

4.1 Datasets and Baselines

To comprehensively evaluate GDAvatar for monocular head avatar reconstruction, we conduct experiments on three representative datasets: INSTA Zielonka et al. (2023), PointAvatar Zheng et al.

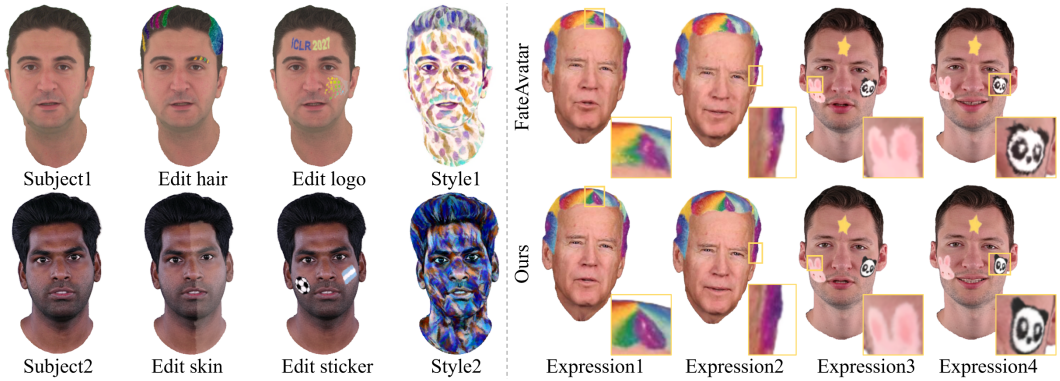

Figure 3: Texture editing results. Our method supports appearance editing, including adding logos and stickers, modifying hair and skin colors, and performing style transfer.

Method	PSNR↑	SSIM↑	LPIPS↓
Geometry			
w/o Laplacian Loss $\mathcal{L}_{lap}$	30.573	0.950	0.049
w/o Geometry Optimization	30.544	0.953	0.051
w/ VHAP Geometry	30.637	0.956	0.048
Appearance			
w/o Dynamic Feature F_d	30.770	0.955	0.050
w/o Attention Fusion	30.689	0.954	0.047
w/o UV Offset Branch	27.913	0.941	0.062
w/ STA UV Offset	30.728	0.952	0.054
w/ Joint Optimization	30.476	0.951	0.058
Full Model	**31.431**	**0.960**	**0.043**

Table 2: Ablation study on the INSTA dataset. Best results are highlighted in bold.

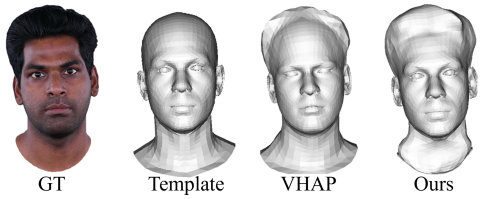

Figure 4: Comparison of meshes.

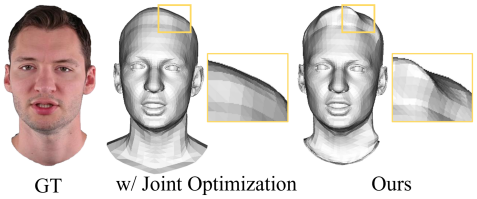

Figure 5: Ablation on Progressive DoF Allocation.

(2023), and NerFace Gafni et al. (2021), and compare with six state-of-the-art Gaussian-based head avatar methods: FlashAvatar Xiang et al. (2023), MonoAvatar Chen et al. (2024), GaussianAvatars Qian et al. (2024), RGBAvatar Li et al. (2025), FateAvatar Zhang et al. (2025), and STAvatar Zhao et al. (2026).

4.2 Implementation Details

All experiments are conducted on a single NVIDIA RTX 4090 GPU. We first perform fast Canonical Geometry Optimization, followed by Dynamic Appearance Modeling, with a learning rate of 1×10^{-4} for both stages. For the Editable Gaussian Map Construction stage, the learning rate is set to 1×10^{-3}. Since monocular videos provide limited observations, we represent Gaussian colors using 0-degree SH. More details, including network configurations, training time, rendering speed, and other implementation settings, are provided in the appendix.

4.3 Experimental Results

Comparison Results. Tab. 1 presents the quantitative comparison on all test datasets. Our method consistently achieves the best performance, demonstrating its effectiveness in reconstructing high-fidelity head avatars. Fig. 2 shows qualitative comparisons on self-reenactment and cross-identity reenactment. Our method reconstructs finer facial details, including wrinkles, eyes, teeth, and hair, benefiting from accurate geometry and the UV Offset Branch, which models dynamic Gaussian

attribute offsets for precise positions and appearances. In contrast, FlashAvatar, GaussianAvatar, and STAvatar suffer from inaccurate facial motions, such as incomplete eye closure. FateAvatar and RGBAvatar produce blurry fine-detail regions, while MonoAvatar exhibits noticeable errors in cross-identity pose transfer.

Geometry Results. For monocular video datasets without ground-truth geometry, existing methods such as GaussianAvatars and STAvatar rely on VHAP Qian (2024) to obtain personalized head geometry through external preprocessing. For a controlled comparison, we evaluate three geometry sources in Stage A: the standard FLAME template, VHAP-estimated geometry, and our optimized geometry. As shown in Fig. 4, the standard FLAME template cannot represent personalized geometry, while the VHAP mesh is relatively coarse, failing to capture fine details such as hair thickness and exhibiting local surface irregularities. In contrast, our optimization produces more detailed and coherent personalized geometry. Their effects on final reconstruction are quantified in Tab. 2: "w/o Geometry Optimization" uses the standard FLAME template, "w/ VHAP Geometry" uses the VHAP-estimated mesh, and the full model uses our optimized geometry. With all other modules unchanged, our method achieves the best reconstruction quality with only about 10 minutes of geometry optimization.

Texture Editing Results. As shown in Fig. 3, our continuous UV attribute maps support various appearance edits, including texture stickers, hair and eyebrow recoloring, skin color editing, and style transfer. Compared with FateAvatar Zhang et al. (2025), our method achieves higher-quality editing while remaining consistent across different poses. Since the continuous UV representation approximates discrete Gaussian attributes, it introduces slight reconstruction degradation; quantitative results are provided in the appendix.

4.4 Ablation Study

Ablation on Geometry Optimization. As shown in Tab. 2, removing Canonical Geometry Optimization degrades reconstruction quality due to less personalized geometry, while removing $\mathcal{L}_{lap}$ causes triangle flipping and reduced local smoothness during optimization.

Ablation on Dynamic Appearance Modeling. The effectiveness of our Dynamic Appearance Modeling mainly comes from the UV Offset Branch. As shown in Tab. 2, removing the Dynamic Feature F_d reduces the fidelity of expression-dependent appearance changes by discarding explicit motion cues. Replacing Attention Fusion with feature concatenation further degrades performance, as fixed fusion cannot adaptively balance canonical identity and dynamic motion features. Without the UV Offset Branch, triangle-bound Gaussians alone cannot capture non-rigid appearance variations. Replacing our UV Offset Branch with the STA UV offset predictor under the same pipeline also yields lower reconstruction quality, demonstrating the benefit of our geometry-conditioned dynamic modeling.

Ablation on Progressive DoF Allocation. We compare our progressive DoF allocation with joint optimization, which simultaneously optimizes geometry and appearance using vertex-bound and triangle-bound Gaussians. As shown in Tab. 2 and Fig. 5, joint optimization performs worse because flexible appearance parameters can compensate for geometric discrepancies, hindering personalized canonical geometry optimization. This result validates the benefit of progressively restricting and releasing Gaussian degrees of freedom rather than jointly optimizing all attributes.

4.5 Conclusion

We propose GDAvatar, a progressive framework that separates personalized canonical geometry from expression-dependent appearance for monocular Gaussian head avatars. By progressively increasing the degrees of freedom of Gaussian primitives, our method first establishes personalized canonical geometry and then models dynamic appearance on this geometric basis. The proposed UV Offset Branch further enables spatially varying Gaussian attributes by adaptively integrating canonical and dynamic information. Moreover, the editable UV representation preserves the learned dynamic variations while providing a continuous interface for appearance editing and style transfer. In future work, we plan to extend our representation to richer geometric structures, including teeth and detailed hair, to further enhance the realism of dynamic head avatars.

AI USE STATEMENT

In this work, we did not use generative AI tools for research ideation, research execution, code development, literature retrieval, manuscript drafting, writing or any other aspect of the research or manuscript preparation. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the network architecture, Gaussian transformations, training procedure, dataset splits, baseline protocols, runtime, and evaluation settings in the main paper and appendix. We will release our code and project resources upon publication.

REFERENCES

Kelian Baert, Mae Younes, Francois Bourel, Marc Christie, and Adnane Boukhayma. Gtavatar: Bridging gaussian splatting and texture mapping for relightable and editable gaussian avatars. In *Computer Graphics Forum*, pp. e70351. Wiley Online Library, 2026.

Shrisha Bharadwaj, Yufeng Zheng, Otmar Hilliges, Michael J Black, and Victoria Fernandez-Abrevaya. Flare: Fast learning of animatable and relightable mesh avatars. *arXiv preprint arXiv:2310.17519*, 2023.

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*, pp. 1–9, 2024.

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 *European Conference on Computer Vision*, pp. 459–476. Springer, 2024.

Wei-Qi Feng, Dong Han, Ze-Kang Zhou, Shunkai Li, Xiaoqiang Liu, Pengfei Wan, Di Zhang, and Miao Wang. Gpavatar: High-fidelity head avatars by learning efficient gaussian projections. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 250–259. IEEE, 2025.

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8649–8658, 2021.

Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 75–82. IEEE, 2018.

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18653–18664, 2022.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Berna Kabadayi, Vanessa Sklyarova, Wojciech Zielonka, Justus Thies, and Gerard Pons-Moll. Physhead: Simulation-ready gaussian head avatars. *arXiv preprint arXiv:2604.06467*, 2026.

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

Linzhou Li, Yumeng Li, Yanlin Weng, Youyi Zheng, and Kun Zhou. Rgbavatar: Reduced gaussian blendshapes for online modeling of head avatars. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10747–10757, 2025.

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, 2017.

Zhanfeng Liao, Yuelang Xu, Zhe Li, Qijing Li, Boyao Zhou, Ruifeng Bai, Di Xu, Hongwen Zhang, and Yebin Liu. Hhavatar: Gaussian head avatar with dynamic hairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Di Liu, Teng Deng, Giljoo Nam, Yu Rong, Stanislav Pidhorskyi, Junxuan Li, Jason Saragih, Dimitris N Metaxas, and Chen Cao. Lucas: Layered universal codec avatars. In *Proceedings of the computer vision and pattern recognition conference*, pp. 21127–21137, 2025.

Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–10, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

SeungJun Moon, Hah Min Lew, Seungeun Lee, Ji-Su Kang, and Gyeong-Moon Park. Geoavatar: Adaptive geometrical gaussian splatting for 3d head avatar. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12811–12821, 2025.

Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12695–12705, 2021.

Sen Peng, Yihang Fu, Runjie Miu, Tianyi Lv, Baorong Yang, and Xiao Dong. Genericavatar: generic human modeling from monocular video based on mesh-guided gaussians: S. peng et al. *The Visual Computer*, 41(9):6657–6670, 2025a.

Sen Peng, Weixing Xie, Zilong Wang, Xiaohu Guo, Zhonggui Chen, Baorong Yang, and Xiao Dong. Rmavatar: Photorealistic human avatar reconstruction from monocular video based on rectified mesh-embedded gaussians. *Graphical Models*, 139:101266, 2025b.

Shenhan Qian. Vhap: Versatile head alignment with adaptive appearance priors. *Zenodo*, 2024.

Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20299–20309, 2024.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

Jack Saunders, Charlie Hewitt, Yanan Jian, Marek Kowalski, Tadas Baltrusaitis, Yiye Chen, Darren Cosker, Virginia Estellers, Nicholas Gydé, Vinay P Namboodiri, et al. Gasp: Gaussian avatars with synthetic priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 271–280, 2025.

Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1606–1616, 2024.

Lei Shi, Sen Peng, Zhiyang Deng, Wenjing Zhang, Rong Jin, and Xiao Dong. Ga-avatar: Geometry-aware human avatar modeling based on gaussian splatting from monocular videos. *IEEE Transactions on Computational Social Systems*, 2026.

Heyi Sun, Cong Wang, Tian-Xing Xu, Jingwei Huang, Di Kang, Chunchao Guo, and Song-Hai Zhang. Svg-head: Hybrid surface-volumetric gaussians for high-fidelity head reconstruction and real-time editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13326–13335, 2025.

Jiapeng Tang, Davide Davoli, Tobias Kirschstein, Liam Schoneveld, and Matthias Niessner. Gaf: Gaussian avatar reconstruction from monocular videos via multi-view diffusion. *arXiv preprint arXiv:2412.10209*, 2024.

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 photorealistic volumetric head avatar. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–12, 2023a.

Cong Wang, Yu-Ping Wang, and Dinesh Manocha. Lolep: Single-view view synthesis with locally-learned planes and self-attention occlusion inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10841–10851, 2023b.

Cong Wang, Di Kang, Heyi Sun, Shenhan Qian, Zixuan Wang, Linchao Bao, and Song-Hai Zhang. Mega: Hybrid mesh-gaussian head avatar for high-fidelity rendering and head editing. In *Proceedings of the computer vision and pattern recognition conference*, pp. 26274–26284, 2025a.

Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. Gaussianhead: High-fidelity head avatars with learnable gaussian derivation. *IEEE Transactions on Visualization and Computer Graphics*, 2025b.

Yating Wang, Xuan Wang, Ran Yi, Yanbo Fan, Jichen Hu, Jingcheng Zhu, and Lizhuang Ma. 3d gaussian head avatars with expressive dynamic appearances by compact tensorial representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21117–21126, 2025c.

Florian Weidner, Gerd Boettcher, Stephanie Arevalo Arboleda, Chenyao Diao, Luljeta Sinani, Christian Kunert, Christoph Gerhardt, Wolfgang Broll, and Alexander Raake. A systematic review on the visualization of avatars and agents in ar & vr displayed using head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2596–2606, 2023.

Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity digital avatar rendering at 300fps. *arXiv preprint arXiv:2312.02214*, 1(2):3, 2023.

Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1931–1941, 2024.

Peizhi Yan, Rabab Ward, Qiang Tang, and Shan Du. Gaussian deja-vu: Creating controllable 3d gaussian head-avatars with enhanced generalization and personalization abilities. *Emotion*, 7712: 6297, 2024.

Haotian Yang, Mingwu Zheng, Wanquan Feng, Haibin Huang, Yu-Kun Lai, Pengfei Wan, Zhongyuan Wang, and Chongyang Ma. Towards practical capture of high-fidelity relightable avatars. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

Jiawei Zhang, Zijian Wu, Zhiyang Liang, Yicheng Gong, Dongfang Hu, Yao Yao, Xun Cao, and Hao Zhu. Fate: Full-head gaussian avatar with textural editing from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5535–5545, 2025.

Jiankuo Zhao, Xiangyu Zhu, Zidu Wang, and Zhen Lei. Stavatar: Soft binding and temporal density control for monocular 3d head avatars reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10996–11005, 2026.

Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13545–13555, 2022.

Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21057–21067, 2023.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9308–9316, 2019.

Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4574–4584, 2023.

A Appendix

A.1 Geometry Optimization Analysis

A.1.1 Geometry Optimization Process

To show the process of the first-stage Canonical Geometry Optimization, we visualize the rendered results and corresponding geometry at different training epochs. As shown in Fig. 6, the geometry quickly approaches the target shape with only a few optimization iterations. By epoch 5, the major head structures have already been recovered, while the remaining optimization mainly focuses on refining local details, such as hair. This convergence is attributed to our vertex-bound Gaussian representation with constrained Gaussian attribute degrees of freedom(DoF), which enables stable and effective geometric adjustment.

A.1.2 Geometry Optimization Strategy

Constrained-DoF. To validate the constrained-DoF design, we conduct ablation studies on opacity and scale. As shown in Fig. 8, learning opacity instead of fixing it to 1 introduces redundant flexibility and leads to inaccurate geometry, especially in the neck region. Similarly, anisotropic scale degrades fine details such as hair. These results demonstrate that constrained-DoF enables more stable geometry optimization.

FLAME Template. We adopt FLAME as the canonical template for its consistent topology, parametric deformation capability, and expression-driven cross-identity animation. The original FLAME template contains approximately 5K vertices and 10K faces. To obtain sufficient geometric details for personalized reconstruction, we further perform one-step mesh subdivision using the subdivision function of PyTorch3D Ravi et al. (2020). The resulting template contains approximately 20K vertices and 40K faces, providing a denser geometric foundation for Gaussian optimization.

A.2 Implementation Details

A.2.1 Dataset Setting

We evaluate our method on three monocular head avatar datasets, including INSTA Zielonka et al. (2023), PointAvatar Zheng et al. (2023), and NerFace Gafni et al. (2021), consisting of 10, 3, and 3 subject sequences, respectively. For data partitioning, we follow the original settings: for the INSTA dataset, we reserve the last 350 frames of each sequence for testing; for the PointAvatar dataset, we adopt the official data split; and for the NerFace dataset, we reserve the last 1,000 frames of each sequence for testing. All input images are resized to a resolution of 512×512. To ensure a fair comparison, we perform quantitative evaluation on the full image regions for all methods, including both the foreground subject and the white background.

A.2.2 Model Details

Progressive Gaussian DoF Allocation. Our method consists of two stages, Canonical Geometry Optimization and Dynamic Appearance Modeling, which progressively allocate Gaussian DoF from geometry modeling to dynamic appearance modeling. As illustrated in Fig. 7, instead of optimizing fully flexible Gaussians from the beginning, we first adopt a vertex-bound Gaussian representation during Canonical Geometry Optimization, where Gaussians are bound to mesh vertices and their DoF are deliberately constrained, reducing them to isotropic spherical Gaussians G_{geo}. By suppressing unnecessary appearance-related flexibility at this stage, persistent reconstruction discrepancies are encouraged to be explained through Gaussian center adjustment rather than absorbed by unconstrained Gaussian attributes. This not only prevents unstable Gaussian drifting, but more importantly enables the optimization to focus on refining the underlying canonical geometry and establishing a personalized geometric basis for subsequent appearance modeling.

Once the canonical geometry is established, we progressively release the Gaussian DoF by rebinding Gaussians to mesh triangles and restoring their full attribute flexibility for Dynamic Appearance Modeling. The base Gaussians G_{base} are constrained on the corresponding mesh surfaces through barycentric coordinates, preserving their consistency with the optimized geometry while

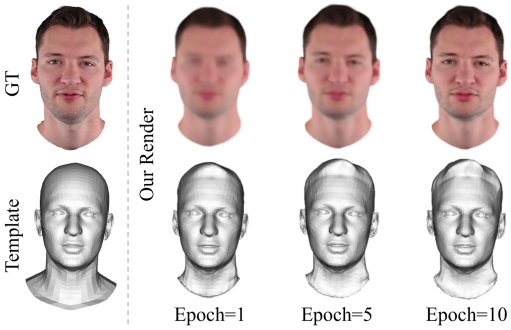

Figure 6: Geometry visualization at different epochs. The overall geometry converges rapidly.

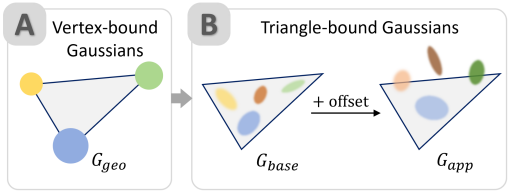

Figure 7: Illustration of our mesh-bound Gaussian representations. Vertex-bound constrained-DoF Gaussians are used for Canonical Geometry Optimization, while triangle-bound full-DoF Gaussians are adopted for Dynamic Appearance Modeling.

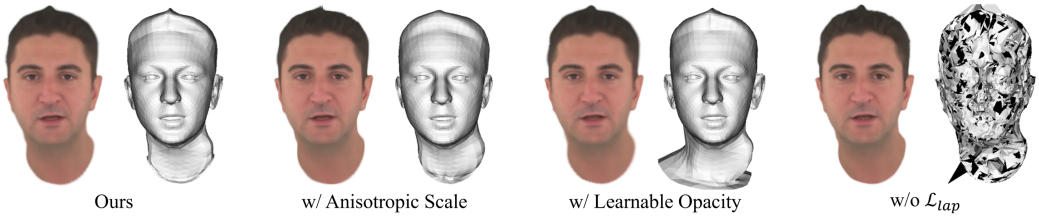

Figure 8: Geometry ablation. Unconstrained Gaussian DoFs degrade geometry quality, while removing the Laplacian loss causes broken mesh structures.

providing denser and more flexible appearance primitives. Since mesh-bound base Gaussians alone are insufficient to represent expression-dependent non-rigid variations, we further learn Gaussian attribute offsets $\Delta\mu, \Delta s, \Delta r, \Delta c, \Delta\sigma$ for position, scale, rotation, color, and opacity, respectively. These offsets are combined with the attributes of the base Gaussian G_{base} to obtain the final dynamic Gaussian representation G_{app} for rendering. The two binding strategies are not independent design choices, but jointly form a progressive Gaussian DoF allocation scheme: constrained vertex-bound Gaussians first establish identity-specific canonical geometry, while triangle-bound Gaussians with full DoF subsequently focus on modeling expression-dependent appearance variations.

Geometry Offset Net. We employ a learnable triplane feature representation for spatial encoding. The triplane is composed of three orthogonal planes with a resolution of 128×128 and 32 feature channels. The sampled features from the three planes are concatenated and processed by a 4-layer MLP Φ_{geo} with a hidden dimension of 128, which predicts the geometric offsets $\Delta\mu_g$ for each vertex.

Latent Encoder. The latent encoder Φ_l is implemented as a U-Net architecture with three downsampling and three upsampling stages. It progressively encodes the input from 64 to 256 feature channels and then decodes the hierarchical features back to a 64-channel representation, producing the latent appearance feature f_{app}.

Canonical Encoder. The canonical encoder Φ_c is implemented as a lightweight 2-layer CNN, consisting of two 3×3 convolution layers with ReLU activations, which produces a 128-channel canonical feature representation F_c.

Dynamic Encoder. We design a dynamic encoder to generate expression-aware dynamic features. A fully connected layer first projects the expression parameters into channel-wise modulation parameters, which are applied after Instance Normalization for expression-conditioned feature modulation. Then, a 3×3 convolution layer predicts the sampling offsets Δp and modulation scalars Δm required by the deformable convolution. With the predicted Δp and Δm, the deformable feature encoder Φ_d with learnable kernel weights and bias terms is employed for dynamic spatial feature modeling. Finally, two convolution layers project the features into a 128-dimensional dynamic feature representation F_d.

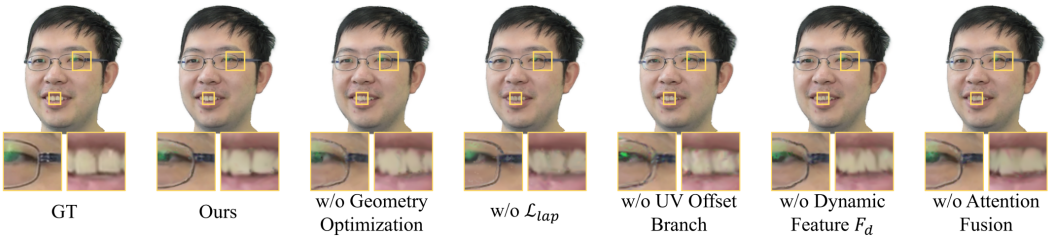

Figure 9: Qualitative ablation results of different components. The full model achieves the best rendering quality.

Attention Fusion. We design an Attention Fusion module to dynamically fuse the Canonical Feature and the Dynamic Feature. Specifically, F_c and F_d are concatenated along the channel dimension and fed into a 2-layer convolutional gating net Φ_{gate} to predict a spatial attention map α. The two features are then adaptively fused according to the predicted α. Finally, the fused feature F_f is passed through another 2-layer convolutional fusion decoder Φ_{fuse} to produce the final UV offset map $O_{off} \in \mathbb{R}^{13\times H\times W}$, which consists of position $O_{off}^{\Delta\mu} \in \mathbb{R}^{3\times H\times W}$, rotation $O_{off}^{\Delta r} \in \mathbb{R}^{3\times H\times W}$, scale $O_{off}^{\Delta s} \in \mathbb{R}^{3\times H\times W}$, color $O_{off}^{\Delta c} \in \mathbb{R}^{3\times H\times W}$, and opacity $O_{off}^{\Delta\sigma} \in \mathbb{R}^{1\times H\times W}$ attribute maps.

Attribute Map Net. The Attribute Map Net adopts a U-Net architecture with a learnable latent feature map $F \in \mathbb{R}^{10\times H\times W}$ as input. The input feature is first projected from 10 to 64 channels through a double convolution block. The encoder then progressively increases the feature channels and reduces the spatial resolution through four downsampling blocks, with the channel dimension increasing from 64 to 512. The decoder restores the spatial resolution through four upsampling blocks and leverages skip connections to fuse multi-scale spatial features, while gradually reducing the feature channels to 64. Finally, a 1×1 convolution layer predicts the 10-channel Gaussian attribute map $O_{base}^{attr} \in \mathbb{R}^{10\times H\times W}$. Since Gaussian positions are inherently defined in the UV space, the position map is not predicted by the network but directly concatenated with the attribute map to form the complete UV base map $O_{base} \in \mathbb{R}^{13\times H\times W}$.

A.2.3 Training Details

All methods are trained on a single NVIDIA RTX 4090 GPU. Following the hyperparameter settings of their original papers, we train FlashAvatar Xiang et al. (2023), MonoAvatar Chen et al. (2024), GaussianAvatars Qian et al. (2024), RGBAvatar Li et al. (2025), FateAvatar Zhang et al. (2025), and STAvatar Zhao et al. (2026) for 20, 100, 10, 10, 10, and 6 epochs, respectively. All baselines are verified to converge. Note that GaussianAvatars is originally designed for multi-view inputs, and we follow its default configuration for evaluation.

For GDAvatar, we adopt a progressive optimization strategy and train the model for 30 epochs, including 10 epochs for geometry and 20 epochs for appearance. During the Canonical Geometry Optimization stage, GeoNet refines the vertex-bound Gaussians with a learning rate of 1×10^{-4} to recover personalized canonical geometry. Then, the Dynamic Appearance Modeling stage employs the UV Offset Branch to learn expression-driven Gaussian attribute offsets with the same learning rate. In both stages, the learning rate is decayed to $1/10$ and $1/100$ at 80% and 90% of the training process, respectively. The learning rates for Gaussian rotation, scale, color, and opacity are set to 1×10^{-3}, 5×10^{-3}, 2.5×10^{-3}, and 5×10^{-2}, respectively. During appearance refinement, we initialize 65K Gaussians uniformly in UV space and apply densification by cloning/splitting every 3K iterations and pruning Gaussians with opacity below 5×10^{-3} every 2K iterations. Opacity reset is not used to ensure stable attribute optimization.

A.3 Additional Results

In this section, we provide additional experimental results, including stage-wise performance, qualitative results, and ablation studies.

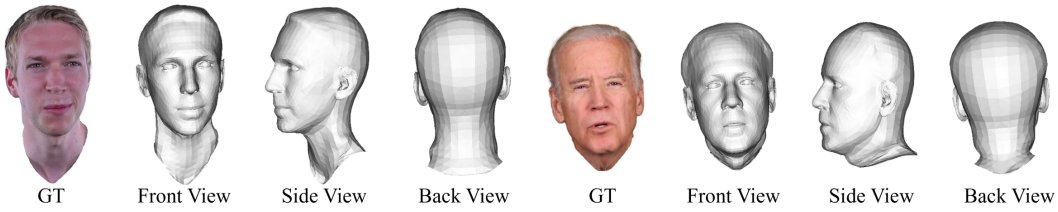

Figure 10: Multi-view visualization of our optimized meshes. Despite limited views in monocular datasets, our method recovers smooth and complete back-head geometry.



Figure 11: Cross-identity reenactment results of our method.

Method	Training Time	FPS	Gaussian Number
FlashAvatar	1.2h	174	16k
MonoAvatar	7.3h	18	100k
GaussianAvatars	0.4h	183	93k
RGBAvatar	0.1h	399	82k
FateAvatar	0.3h	186	55k
STAvatar	0.5h	62	71k
Ours	0.8h	79	49k

Table 3: Comparison of training time, rendering speed, and Gaussian number on the "bala" subject.

A.3.1 Performance of Different Stages

As shown in Tab. 4, the vertex-bound constrained-DoF Gaussian representation achieves lower quantitative performance. Therefore, we further adopt the triangle-bound full-DoF Gaussian representation for appearance modeling, achieving the best reconstruction quality. For controllable editing, we further convert the discrete Gaussian attributes into a continuous UV representation. This improves spatial correspondence and editability, at the cost of a slight decrease in reconstruction quality.

A.3.2 Optimized Mesh

Fig. 10 visualizes the optimized meshes produced by our method. The results show that our optimized geometry captures subject-specific characteristics, including personalized hair structures and neck lengths. Despite being optimized under limited-view supervision, the reconstructed meshes maintain smooth and plausible backside geometry without unnatural depressions.

A.3.3 Cross-identity Reenactment Results

Fig. 11 presents more cross-identity reenactment results of our method. Our approach effectively transfers facial motions across different identities, including head pose, mouth articulation, and eye movements, while preserving identity-specific appearance.

A.3.4 Editing Results

Fig. 12 demonstrates the editing results on different identities. Our editable Gaussian representation enables flexible appearance manipulation, including adding text and stickers, changing hair and skin colors, and applying style transfer effects. The corresponding reconstructed UV color texture maps are also visualized, which are directly modified during editing to achieve the desired appearance changes.



Figure 12: Appearance editing results on different identities.

Stage	INSTA			PointAvatar			NerFace		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Canonical Geometry Optimization	25.675	0.921	0.152	23.302	0.887	0.208	26.549	0.924	0.147
Dynamic Appearance Modeling	31.431	0.960	0.043	29.184	0.936	0.063	34.162	0.975	0.021
Editable Gaussian Map Construction	30.719	0.946	0.050	28.657	0.931	0.074	32.736	0.963	0.038

Table 4: Reconstruction performance of different stages on the INSTA, PointAvatar, and NerFace datasets.

A.3.5 Ablation Results

Fig. 9 provides additional qualitative results of our ablation studies. The full model achieves the best rendering quality with more complete facial details. Without Geometry Optimization, the model directly uses the FLAME template (Fig. 6) for Gaussian binding, resulting in less personalized geometry. Removing the Laplacian loss leads to broken mesh structures (Fig. 8), which further degrade the rendering quality. For appearance modeling, removing the UV Offset Branch constrains Gaussians to the mesh surface and prevents local attribute refinement beyond the mesh geometry. Without the Dynamic Feature, the model cannot effectively capture expression-dependent appearance variations. Meanwhile, removing the Attention Fusion limits the integration of canonical priors and dynamic motion information, resulting in less accurate appearance reconstruction.

A.4 Computational Efficiency Analysis

We further compare the computational efficiency of different methods in Tab. 3. Our method uses only 49K Gaussians, fewer than most Gaussian-based approaches, benefiting from the accurate geometry for Gaussian binding. Although our training time is slightly longer due to the additional geometry optimization and dynamic appearance modeling stages, our method achieves 79 FPS during inference, including both animation and rendering time. Overall, our method achieves a good trade-off between reconstruction quality, model complexity, and rendering efficiency.

A.5 Limitations

The original FLAME template lacks detailed internal oral structures. We augment the template with additional faces to model the oral cavity, following similar strategies adopted in FlashAvatar, enabling basic oral structure representation. In the future, we will explore more flexible geometric optimization methods to enhance modeling accuracy for the hair, mouth, and neck regions.