

# HRAvatar: High-Quality and Relightable Gaussian Head Avatar

## Supplementary Material

### Overview

This supplementary material presents more details and additional results not included in the main paper due to page limitation. The list of items included are:

- A video demo (HRAvatar\_video\_demo.mp4) with a brief description of the video results in Appendix A.
- More model implementation details in Appendix B.
- Additional comparison with FLARE and ablation study in Appendix C.
- Application results for novel view synthesis and material editing in Appendix D.
- Further discussion on method differences, limitations, and ethical considerations in Appendix E.

### A. Video Demo

We strongly encourage readers to watch the video provided in the supplementary materials or on the [project page](#). It showcases the self-reenactment animation of avatars reconstructed by HRAvatar and includes novel view renderings. The video also illustrates the visual results of relighting the avatars under various rotating environment maps and the ability to perform simple material editing to enhance specular reflections. Furthermore, we provide visual comparisons of HRAvatar with two advanced methods, GBS [8] and Flash-avatar [11], in self-reenactment, cross-reenactment, and novel view synthesis. A relighting comparison with FLARE [1] is also included. Overall, the video highlights our method’s capability to create fine-grained avatars with excellent expressiveness and realistic lighting effects in diverse environments.

### B. More Implementation Details

#### B.1. Preliminary

3D Gaussian Splatting [6] represents 3D scene with explicit Gaussian points, each point  $G$  is defined by its position (center)  $X$ , rotation  $r$ , scaling  $s$ , opacity  $\alpha$  and color  $c$ . During rendering, each Gaussian point affects nearby pixels anisotropically using a Gaussian function  $\mathcal{G}$ :

$$\mathcal{G}(x, \mu', \Sigma_{2D}) = e^{-\frac{1}{2}(x-\mu')^\top \Sigma_{2D}^{-1}(x-\mu')}, \quad (1)$$

where  $\mu'$  is the projected mean of  $X$  on the image plane. Given the viewing transformation  $W$ , the 2D covariance matrix  $\Sigma_{2D}$  is derived from the 3D covariance matrix:

$$\Sigma_{2D} = JW\Sigma W^\top J^\top, \Sigma = RSS^\top R^\top. \quad (2)$$

$J$  is the Jacobian of the affine approximation of the projective transformation. To ensure the covariance matrix  $\Sigma$








	Rendering Quality	Relighting	Rendering speed
Point-Avatar [12]	 0.646	Limited	≈ 6 FPS
INSTA [13]	 0.764	✗	≈ 1 FPS
FLARE [1]	 0.698	✓	≈ 35 FPS
Splatting-avatar [10]	 0.834	✗	> 120 FPS
Flash-avatar [11]	 0.883	✗	> 120 FPS
GBS [8]	 0.980	✗	> 120 FPS
HRAvatar (Ours)	 1.184	✓	> 120 FPS

Table 1. Key aspects of our method compared to previous works. The rendering quality shows the inverse of the MAE metric on the INSTA dataset, with longer bars representing better performance. ‘Limited’ indicates that the Point-Avatar method has limited flexibility in handling relighting.

remains positive semi-definite during optimization, it is decomposed into a scaling matrix  $S$  and a rotation matrix  $R$ , as Eq. (2). The scaling matrix  $S$  and rotation matrix  $R$  are represented by a 3D vector  $s$  and a quaternion  $r$ , respectively. The color  $c$  is modeled by a third-order spherical harmonic coefficient for view-dependent effects. During splatting, the image space is divided into multiple  $16 \times 16$  tiles, and pixel colors are computed with alpha blending:

$$\mathcal{C}(x_p) = \sum_{i \in G_{x_p}} c_i \sigma_i \prod_{j=1}^{i-1} (1 - \sigma_j), \sigma_i = \mathcal{G}(x_p, \mu'_i, \Sigma_{2D,i}) \alpha_i, \quad (3)$$

where,  $x_p$  represents the pixel position, and  $G_{x_p}$  denotes the sorted Gaussian points associated with pixel  $x_p$ . Additionally, a strategy is proposed to adjust the number of Gaussian points through densification and pruning.

#### B.2. Training Details

In the first 1500 iterations, we take the albedo map as the rendered image to learn the head’s albedo properties initially. Afterward, we switch to shaded image to learn other attributes. Each Gaussian point’s roughness, Fresnel base reflectance, and albedo attributes are initialized to 0.9, 0.04, and 0.5, respectively. While we generally follow 3DGS hyperparameters, we make some adjustments. During training, point densification starts at iteration 1000 and ends at 500 iterations before training completes, with a densification interval of 500 iterations. The gradient threshold is increased to  $3 \times 10^{-4}$  to avoid excessive point growth. During training, opacity is reset below the pruning threshold to eliminate more redundant points. The learning rates for the Gaussian point positions, appearance attributes, and environment map gradually decrease as training progresses, while the expression encoder learning rate is set to  $5 \times 10^{-5}$ .

Training a video with 2400 frames takes about one hour.

When using albedo prior to supervision, we apply it every 3 frames due to the time-consuming process of extracting pseudo-ground-truth albedo during preprocessing. Additionally, since the lighting in the INSTA and self-captured datasets is relatively uniform, we only apply albedo prior supervision during training on the HDTF dataset. Furthermore, for subjects in the HDTF dataset, we set a higher upper bound for reflectance ( $\tau_{max}^{f_0}$ ) to account for the specific lighting conditions.

### B.3. Model Details

The shape and expression basis in FLAME are derived through PCA, with higher dimensions having a small effect on deformation. To avoid unnecessary computations, we use only the first 100 shape parameters and 50 expression parameters, i.e.,  $|\beta| = 100$  and  $|\psi| = 50$ . Since FLAME lacks an interior mesh for the mouth, we follow Qian et al. [9] by adding a mesh for the teeth, where the upper and lower teeth move according to the neck and jaw joints, respectively. Additionally, we add extra mesh behind the teeth to provide a reasonable initialization for the rest of the mouth interior.

During shading, normal and reflection vectors sample lighting from the irradiance and pre-filtered environment maps. Since both maps must be backpropagated and mipmaps reconstructed in each training iteration, the computation increases with resolution. To maintain efficient training, we set the irradiance map  $I_{irr}$  resolution to  $16 \times 16$  and the pre-filtered environment map  $I_{env}$  to  $32 \times 32$  with 3 mipmap levels.

### B.4. BRDF Reflection Model.

For physical-based shading, we use the Disney model [2] to describe light interactions with geometry and materials, a method commonly employed in real-time rendering. This model breaks reflection into two components: Lambertian diffuse reflection and specular reflection:

$$L_o(X, \omega_o) = L_d + L_s = \int_{\Omega} \frac{a}{\pi} L_i(X, \omega_i) n \cdot \omega_i d\omega_i + \int_{\Omega} \frac{\mathcal{DFH}}{4(n \cdot \omega_o)(n \cdot \omega_i)} L_i(X, \omega_i) n \cdot \omega_i d\omega_i, \quad (4)$$

where  $L_i$  and  $L_o$  denote the radiance for the incoming direction  $\omega_i$  and outgoing direction  $\omega_o$ , respectively with  $n$  as the normal. The Lambertian term models diffuse reflection, independent of viewing direction, allowing us to precompute and store this part in an irradiance map. The specular reflection term models appearance based on viewing angle, with  $\mathcal{D}$ ,  $\mathcal{F}$ , and  $\mathcal{H}$  representing the normal distribution, Fresnel equation, and geometric function. We use the Split-Sum approximation [5] to simplify the BRDF integral into

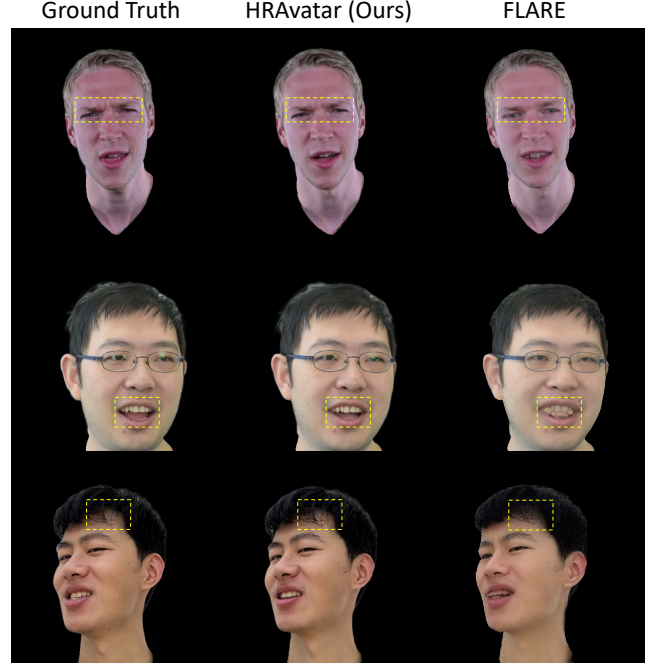


Figure 1. Visual comparison with FLARE on self-reenactment. Our method captures facial expression details more effectively and reconstructs the teeth geometry and hair texture more accurately.

two parts:

$$L_s \approx I_{env} \cdot I_{BRDF} = \left( \frac{1}{Z} \sum_{z=1}^Z L_i(\omega_z) \right) \cdot \left( \frac{1}{Z} \sum_{z=1}^Z \frac{\mathcal{DFH} \cdot n \cdot \omega_z}{4(n \cdot \omega_o)(n \cdot \omega_z) pdf(\omega_z, \omega_o)} \right). \quad (5)$$

Here,  $pdf(\omega_m, \omega_o)$  is the probability density function related to  $\mathcal{D}$ . Both components are precomputed and stored:  $I_{env}$  as a multi-resolution mipmap for different roughness levels and  $I_{BRDF}$ , as a lookup table (LUT) based on roughness and the dot product of the normal and observation direction,  $n \cdot \omega_o$ .

## C. Further Experiments

### C.1. Rendering Speed

Despite the additional computational load introduced by the deformation and appearance models, our method still achieves real-time rendering speeds. To provide a reference, we test the rendering speed on the INSTA dataset using a single NVIDIA 3090 GPU. Each trained avatar contains about 75K Gaussian points. We set the rendering resolution to  $512 \times 512$  and render 500 images to calculate the average speed. HRAvatar achieves an average speed of about **155 FPS**, with the encoder extracting parameters at about 179

Method	INSTA dataset				HDTF dataset				self-captured dataset			
	PSNR $\uparrow$	MAE $\downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	MAE $\downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	MAE $\downarrow$	SSIM $\uparrow$	LPIPS $\downarrow$
FLARE	26.80	1.433	0.9063	0.0816	25.55	2.193	0.8479	0.1183	25.82	1.715	0.8576	0.1230
HRAvatar (Ours)	<b>30.36</b>	<b>0.845</b>	<b>0.9482</b>	<b>0.0569</b>	<b>28.55</b>	<b>1.373</b>	<b>0.9089</b>	<b>0.0825</b>	<b>28.97</b>	<b>1.123</b>	<b>0.9054</b>	<b>0.1059</b>

Table 2. Average quantitative results on the INSTA, HDTF, and self-captured datasets. Our method outperforms FLARE in PSNR, MAE\* (MAE  $\times 10^2$ ), SSIM, and LPIPS metrics.

	albedo (LMSE $\downarrow$ )	normal (cosine similarity $\uparrow$ )
FLARE	0.0665	0.8424
Ours	<b>0.0557</b>	<b>0.9093</b>

Table 3. Albedo and normal evaluation on the HDTF Dataset.

FPS. Similarly, when **relighting** with a new environment map, we measured a rendering speed of approximately **155 FPS** under the same setup, ensuring real-time performance.

## C.2. Comparison with FLARE

Since both FLARE [1] and our method can perform monocular 3D head reconstruction and relighting, we conduct a further comparison.

**Self-reenactment.** The experimental setup is the same as in the main paper, with quantitative results shown in Tab. 2 and qualitative results in Fig. 1. Our method outperforms FLARE in both metrics and visual quality, better capturing details of facial expressions, hair textures, and internal mouth features such as teeth.

**Speed.** Under the same setup, we test FLARE’s average rendering speed on the INSTA dataset, which is approximately 35 FPS. In contrast, our method achieves a rendering speed of about **4.5 $\times$**  higher.

**Disentanglement and geometric.** Directly evaluating material disentanglement is challenging due to the scarcity of publicly available real or synthetic face video datasets. As an alternative, we employ SwitchLight [7] to extract image albedo as pseudo-ground truth for evaluation. We compare against FLARE using LMSE (Local Mean Squared Error) [4] as the evaluation metric. Results are in Tab. 3. Roughness and reflectance are excluded due to varying definitions and usage across shading models.

Normals are commonly used to assess reconstructed 3D geometry. To quantify this, since we lack ground truth normals, we use the SOTA single-image geometry estimation method, GeoWizard [3], to estimate normals from the images as pseudo-ground truth. We use the cosine similarity of normals as the evaluation metric, with results shown in Tab. 3.

The qualitative comparison of normals and decoupling results is shown in Fig. 4 of the main paper.

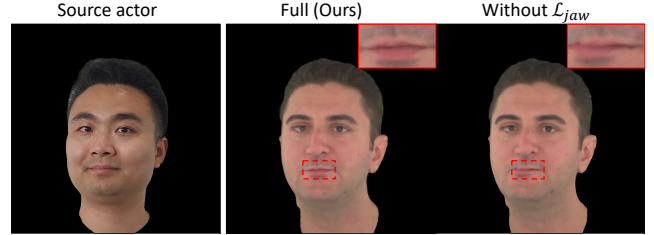


Figure 2. Ablation result on  $\mathcal{L}_{jaw}$ . Without the jaw pose regularization loss, the avatar exhibits mouth distortion during cross-reenactment.

## C.3. Ablation Of Jaw Pose Regularization Loss

Without the jaw pose regularization loss,  $\mathcal{L}_{jaw}$ , the trained encoder may extract jaw poses that deviate from the normal distribution. This can lead to incorrect mouth motion during cross-reenactment. As shown in Fig. 2, removing  $\mathcal{L}_{jaw}$  results in mouth distortion, while including this loss effectively prevents the issue.

## C.4. Complete Quantitative Results

We present the complete quantitative results of self-reenactment for each subject on the INSTA, HDTF, and self-captured datasets in Tab. 4 and Tab. 5. As shown, HRAvatar achieves superior performance for most subjects, demonstrating the robustness of our method.

## D. Applications

### D.1. Relighting

We show the relighting results of the head illuminated by rotating environment maps in Fig. 3. For each map, we extract the corresponding irradiance and prefiltered maps, applying them in the shading process (Sec. 3.3). HRAvatar achieves real-time rendering speed during relighting

For convenience during relighting, we use off-the-shelf tools to precompute the irradiance map and pre-filtered environment map from the environment map. Specifically, we use **CmftStudio**, a tool commonly used in real-time rendering pipelines to process HDR images for image-based lighting. With CmftStudio, we extract the original environment map with a resolution of  $1024 \times 512$  into an irradiance map of  $512 \times 256$  and a pre-filtered environment map with 7 mipmaps, ranging from  $1024 \times 512$  to  $16 \times 8$ .

		INSTA dataset									
		bala	biden	justin	malte.1	marcel	nf.01	nf.03	obama	person0004	wojteck.1
PSNR $\uparrow$	INSTA	29.53	29.92	31.66	27.44	22.99	26.45	28.31	31.21	25.44	31.36
	Point-avatar	27.88	27.64	30.40	24.98	24.66	25.25	26.60	28.83	23.29	28.82
	FLARE	27.20	28.55	29.10	25.93	22.50	25.97	26.71	28.67	25.53	27.84
	Splatting-avatar	32.14	30.42	30.93	27.66	24.34	27.08	27.85	30.64	<u>26.49</u>	29.54
	Flash-avatar	30.27	31.25	32.16	27.45	24.85	<u>28.02</u>	<u>28.28</u>	<u>31.46</u>	25.49	<u>32.03</u>
	GBS	<u>32.47</u>	<b>32.23</b>	<u>33.10</u>	<u>28.23</u>	<u>26.11</u>	27.59	28.12	31.35	25.16	<b>32.05</b>
	HRAvatar (Ours)	<b>33.10</b>	<u>31.70</u>	<b>33.29</b>	<b>29.28</b>	<b>26.58</b>	<b>28.95</b>	<b>29.68</b>	<b>33.24</b>	<b>26.54</b>	31.26
MAE* $\downarrow$	INSTA	1.154	0.849	0.642	1.160	2.996	1.705	1.381	0.775	1.594	0.834
	Point-avatar	1.386	1.203	0.869	1.596	2.662	1.800	1.583	1.103	2.083	1.042
	FLARE	1.342	0.973	0.910	1.470	2.817	1.706	1.602	1.097	1.392	1.020
	Splatting-avatar	0.854	0.838	0.783	1.135	2.309	1.533	1.340	0.917	<u>1.376</u>	0.910
	Flash-avatar	1.175	0.670	0.610	1.058	2.133	1.326	1.249	0.819	1.589	0.700
	GBS	<u>0.747</u>	<b>0.583</b>	<u>0.520</u>	<u>1.010</u>	<u>1.608</u>	<u>1.311</u>	<u>1.162</u>	<u>0.802</u>	1.803	<b>0.655</b>
	HRAvatar (Ours)	<b>0.657</b>	<u>0.616</u>	<b>0.498</b>	<b>0.902</b>	<b>1.293</b>	<b>1.133</b>	<b>1.031</b>	<b>0.580</b>	<b>1.070</b>	<u>0.668</u>
SSIM $\uparrow$	INSTA	0.8896	0.9460	0.9591	0.9159	0.8736	0.8937	0.8676	0.9484	0.8478	0.9452
	Point-avatar	0.8658	0.9116	0.9373	0.8853	0.9063	0.8919	0.8807	0.9145	0.8576	0.9192
	FLARE	0.8761	0.9347	0.9363	0.8973	0.8892	0.9027	0.8841	0.9199	0.9015	0.9216
	Splatting-avatar	0.9272	0.9466	0.9482	0.9243	0.9041	0.9202	0.9113	0.9411	<u>0.9075</u>	0.9400
	Flash-avatar	0.8494	0.9614	0.9611	0.9326	0.9086	0.9270	0.9155	<u>0.9493</u>	0.8996	0.9509
	GBS	<u>0.9390</u>	<b>0.9658</b>	0.9690	<u>0.9374</u>	<u>0.9217</u>	<u>0.9365</u>	<u>0.9271</u>	0.9476	0.8910	<b>0.9593</b>
	HRAvatar (Ours)	<b>0.9473</b>	<u>0.9635</u>	<u>0.9687</u>	<b>0.9429</b>	<b>0.9352</b>	<b>0.9398</b>	<b>0.9334</b>	<b>0.9647</b>	<b>0.9278</b>	<u>0.9590</u>
LPIPS $\downarrow$	INSTA	0.0992	0.0541	0.0521	0.0731	0.1351	0.1262	0.1286	0.0446	0.1453	0.0540
	Point-avatar	0.0829	0.0637	0.0588	0.0758	0.1247	0.1257	0.1143	0.0589	0.1637	0.0576
	FLARE	0.0927	0.0513	0.0582	0.0726	0.1266	0.1068	0.0971	0.0595	<u>0.0947</u>	0.0567
	Splatting-avatar	0.0865	0.0564	0.0651	0.0749	0.1326	0.1107	0.0966	0.0545	0.1246	0.0602
	Flash-avatar	0.1535	0.0299	<u>0.0378</u>	<u>0.0477</u>	<u>0.1069</u>	<u>0.0868</u>	<u>0.0760</u>	<u>0.0376</u>	0.1035	<u>0.0392</u>
	GBS	<u>0.0862</u>	0.0433	0.0481	0.0737	0.1219	0.1076	0.0861	0.0564	0.1417	0.0582
	HRAvatar (Ours)	<b>0.0451</b>	<u>0.0306</u>	<b>0.0367</b>	<b>0.0476</b>	<b>0.0992</b>	<b>0.0868</b>	<b>0.0649</b>	<b>0.0279</b>	<b>0.0940</b>	<b>0.0358</b>

Table 4. Complete quantitative results of self-reenactment for each subject on the INSTA dataset. HRAvatar achieves better performance metrics in most cases. Bold marks the best, and underline marks the second.

		HDTF dataset								self-captured dataset				
		elijah	haaland	katie	marcia	randpaul	schako	tom	veronica	s1	s2	s3	s4	s5
PSNR $\uparrow$	INSTA	25.00	24.94	21.36	24.61	23.50	26.45	29.16	26.45	25.88	25.37	29.33	24.86	24.086
	Point-avatar	24.05	25.56	22.51	23.76	26.28	25.44	27.01	26.51	25.35	27.32	28.09	23.56	24.85
	FLARE	25.05	25.66	22.10	23.58	26.98	25.05	29.45	26.50	26.26	26.12	28.32	24.07	24.32
	Splatting-avatar	26.08	26.31	22.23	25.80	29.25	25.51	30.98	27.14	25.05	28.20	29.54	25.34	24.22
	Flash-avatar	26.29	26.46	<u>23.39</u>	<u>26.67</u>	29.05	<b>28.28</b>	<u>31.56</u>	<u>28.95</u>	26.37	27.26	30.59	<b>28.01</b>	25.09
	GBS	<u>26.76</u>	<u>28.29</u>	22.74	26.59	<u>29.20</u>	27.88	31.54	29.48	<u>28.15</u>	<u>29.50</u>	<b>31.64</b>	<u>27.48</u>	<u>26.17</u>
	HRAvatar (Ours)	<b>28.24</b>	<b>28.91</b>	<b>24.92</b>	<b>27.23</b>	<b>29.70</b>	<u>27.95</u>	<b>31.75</b>	<b>29.71</b>	<b>29.40</b>	<b>30.19</b>	<u>31.40</u>	27.00	<b>26.84</b>
MAE* $\downarrow$	INSTA	1.835	2.161	4.179	2.191	2.602	1.936	1.272	2.487	1.877	1.637	1.377	1.841	2.807
	Point-avatar	2.058	2.177	3.493	2.423	1.746	2.092	1.683	2.212	1.852	1.312	1.204	1.903	2.210
	FLARE	1.813	2.097	3.732	2.580	1.637	2.207	1.204	2.277	1.762	1.540	1.209	1.736	2.328
	Splatting-avatar	1.652	1.915	3.841	2.026	1.260	2.200	0.988	2.183	2.093	1.296	1.110	1.565	2.489
	Flash-avatar	1.602	2.052	<u>2.922</u>	1.755	1.312	1.519	0.980	1.865	1.909	1.364	1.079	<u>1.251</u>	2.557
	GBS	<u>1.406</u>	<u>1.403</u>	3.216	<u>1.659</u>	<u>1.234</u>	<u>1.452</u>	<u>0.901</u>	<u>1.535</u>	<u>1.379</u>	<u>1.022</u>	0.950	1.285	<u>2.018</u>
	HRAvatar (Ours)	<b>1.108</b>	<b>1.319</b>	<b>2.283</b>	<b>1.483</b>	<b>1.079</b>	1.384	<b>0.847</b>	<b>1.477</b>	<b>1.142</b>	<b>0.896</b>	<b>0.792</b>	<b>1.117</b>	<b>1.666</b>
SSIM $\uparrow$	INSTA	0.8808	0.8337	0.7474	0.8290	0.8528	0.8586	0.9143	0.7700	0.8218	0.8659	0.8722	0.8634	0.7431
	Point-avatar	0.8631	0.8275	0.7771	0.8160	0.8694	0.8578	0.8634	0.8339	0.8460	0.8763	0.8867	0.8573	0.8117
	FLARE	0.8798	0.8426	0.7773	0.8117	0.8773	0.8517	0.9064	0.8364	0.8522	0.8560	0.8878	0.8716	0.8204
	Splatting-avatar	0.8952	0.8562	0.7562	0.8477	0.9094	0.8586	0.9321	0.8337	0.8279	0.8775	0.9038	0.8817	0.8031
	Flash-avatar	0.8898	0.8146	0.8133	0.8636	0.9040	0.8982	0.9305	0.8170	0.7774	0.8659	0.8967	0.8850	0.7491
	GBS	0.9113	0.8924	0.8068	<u>0.8783</u>	0.9110	0.9091	0.9404	<u>0.8826</u>	0.8799	0.9098	0.9188	0.9029	0.8339
	HRAvatar (Ours)	<b>0.9335</b>	<b>0.9036</b>	<b>0.8597</b>	<b>0.8961</b>	<b>0.9254</b>	<b>0.9135</b>	<b>0.9446</b>	<b>0.8951</b>	<b>0.9019</b>	<b>0.9232</b>	<b>0.9283</b>	<b>0.9142</b>	<b>0.8596</b>
LPIPS $\downarrow$	INSTA	0.1005	0.1698	0.2222	0.1586	0.1417	0.1390	0.0729	0.2415	0.1897	0.1583	0.1523	0.1678	0.2483
	Point-avatar	0.0886	0.1360	0.1683	0.1200	0.1147	0.1283	0.0981	0.1686	0.1255	0.0942	0.1024	0.1364	0.1623
	FLARE	0.0821	0.1255	0.1589	0.1258	0.1040	0.1193	0.0748	0.1559	0.1217	0.1014	0.1088	0.1331	<u>0.1500</u>
	Splatting-avatar	0.0902	0.1476	0.1982	0.1385	0.1033	0.1455	0.0664	0.1907	0.1773	0.1271	0.1194	0.1539	0.1972
	Flash-avatar	<u>0.0759</u>	0.1595	<u>0.1387</u>	<u>0.0881</u>	<u>0.0829</u>	<u>0.1011</u>	<u>0.0609</u>	<u>0.1688</u>	0.2346	<u>0.0736</u>	<b>0.0901</b>	<b>0.109</b>	0.2208
	GBS	0.0875	<u>0.1515</u>	0.1899	0.1289	0.1113	0.1160	0.0679	0.1850	<u>0.1696</u>	0.1198	0.1305	0.1599	0.2004
	HRAvatar (Ours)	<b>0.0504</b>	<b>0.0929</b>	<b>0.1208</b>	<b>0.0723</b>	<b>0.0683</b>	<b>0.0846</b>	<b>0.0485</b>	<b>0.12228</b>	<b>0.1063</b>	<b>0.0662</b>	<u>0.0939</u>	<u>0.1153</u>	<b>0.1478</b>

Table 5. Complete quantitative results of self-reenactment for each subject on the HDTF and self-captured dataset. HRAvatar achieves better performance metrics in most cases.

## D.2. Material Editing

By modeling the avatar’s material properties for physical shading, we can easily edit the avatar’s materials. In Fig. 4, we show material editing under new lighting conditions by gradually increasing the base Fresnel reflectance, which enhances the metallic effect and reduces diffuse reflection. As

shown, higher reflectance results in stronger specular reflections, validating the effectiveness of our physically-based shading model.

## D.3. Novel Views Synthesis

Although the 3D avatar is reconstructed from a monocular video, it can still render novel views. Fig. 5 shows the vi-



Reconstruct

Relighting by rotating light

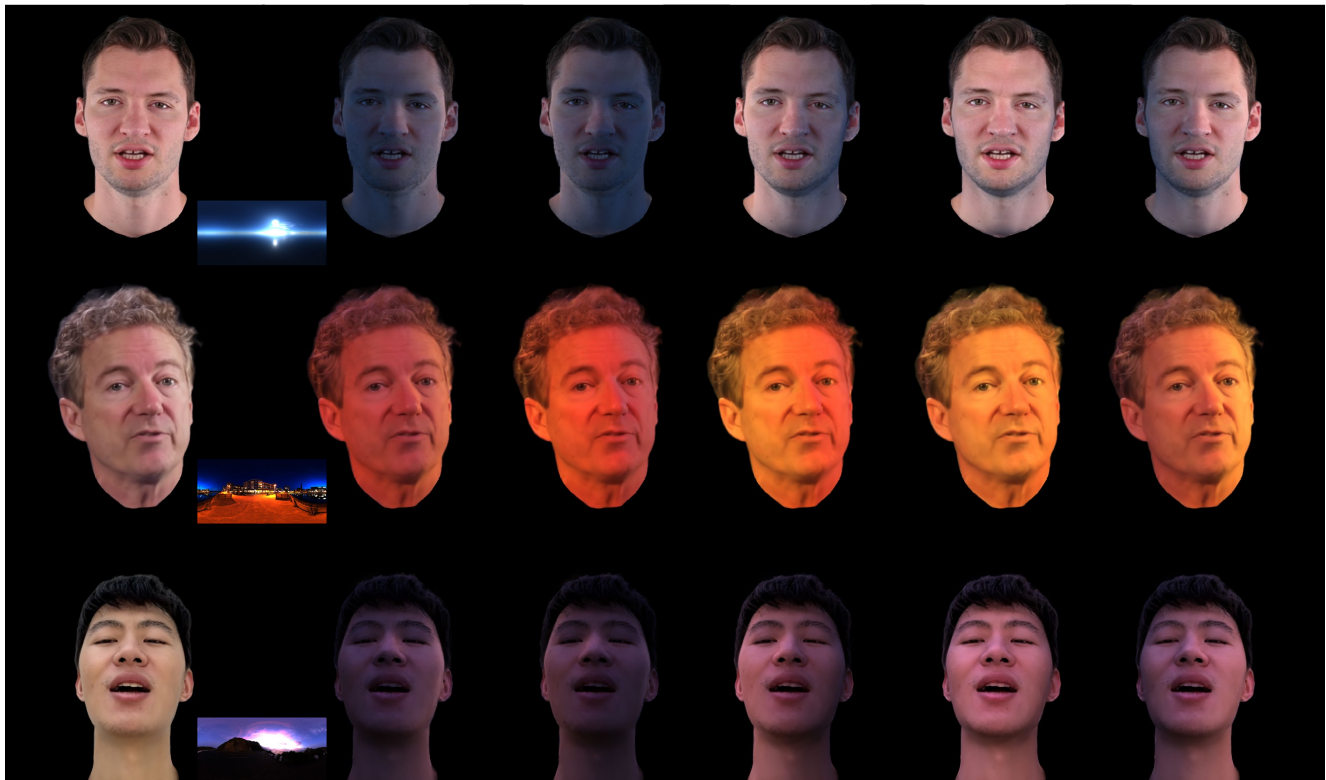


Figure 3. Relighting visual results. For each environment map, we rotate the lighting to illuminate the head from different directions.

Reconstruct

Material Editing with increasing base fresnel reflectance



Figure 4. Visual results of material editing. We gradually increase the avatar's base Fresnel reflectance under new environment lighting, enhancing specular reflections. The results align with intuitive expectations, validating the effectiveness of our shading model.

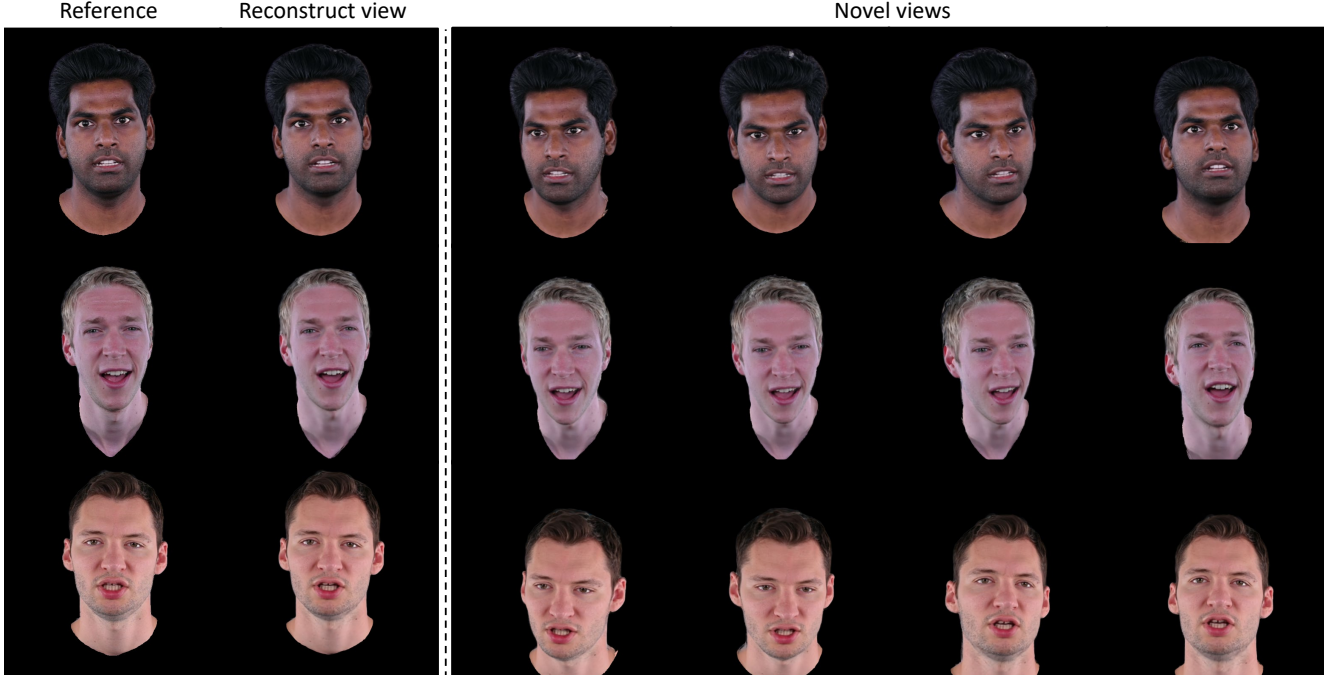


Figure 5. Visual results of novel view synthesis. In each row, the original view of the reconstructed subject is shown on the left, while the rendered novel views are on the right. Our method produces high-fidelity novel views with strong 3D consistency.

sual results of our method. As shown, HRAvatar renders novel views of the head with high 3D consistency and quality, preserving fine texture details.

## E. More Discussion

### E.1. Method Comparison

**FLARE.** Similar to most relighting methods, both FLARE and our approach use a BRDF reflection model to account for environmental lighting on head appearance. The key distinction lies in the 3D representation: FLARE adopts a mesh-based approach, while we leverage 3D Gaussian Splatting (3DGS) and extend it with physically-based shading. We further overcome 3DGS’s limitations in modeling normals and decoupling highlights from albedo. Moreover, our improved deformation model further enables higher-fidelity avatar reconstruction while achieving faster rendering compared to FLARE.

**3DGS-based. GBS.** While both GBS and our method employ blendshapes to model positional displacements, we introduce: 1) learnable blend skinning for per-point rotations; 2) end-to-end training of an expression encoder to enhance tracking; and 3) a novel appearance model for better material decomposition and relighting. *Other 3DGS-based.* Compared to other existing 3DGS-based monocular reconstruction methods, HRAvatar introduces a more flexible deformation method and employs an end-to-end trained expression encoder for more accurate expression capture,

leading to superior reconstruction quality. Furthermore, we pioneer realistic, relightable monocular Gaussian head reconstruction. The main differences are summarized in Tab. 1.

### E.2. Future improvements.

The extra computation from blendshapes, linear skinning, and shading slows down 3DGS rendering, but offloading these tasks to the GPU via CUDA could mitigate this. These challenges present opportunities for future improvement.

While albedo supervision from existing models reduces coupling to some extent, highlights may still be misattributed to properties like roughness or reflectance. Ideally, the same region, such as hair or skin, should have consistent material attributes. Introducing semantic information to guide and constrain material learning is a promising future direction.

### E.3. Ethical Considerations.

Creating realistic, controllable head avatars raises concerns about potential violations of portrait rights and privacy. It may also lead to identity theft and misuse in fraud. We strongly condemn any unauthorized use of this technology for illegal purposes. It’s crucial to consider ethical implications in all applications of our method to prevent harm to the public.

## References

- [1] 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. [1](#), [3](#)
- [2] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm Siggraph*, pages 1–7. vol. 2012, 2012. [2](#)
- [3] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024. [3](#)
- [4] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2335–2342. IEEE, 2009. [3](#)
- [5] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. [2](#)
- [6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. [1](#)
- [7] Hoon Kim, Minje Jang, Wonjun Yoon, Jisoo Lee, Donghyun Na, and Sanghyun Woo. Switchlight: Co-design of physics-driven architecture and pre-training framework for human portrait relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25096–25106, 2024. [3](#)
- [8] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024. [1](#)
- [9] 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*, pages 20299–20309, 2024. [2](#)
- [10] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [1](#)
- [11] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity head avatar with efficient gaussian embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [1](#)
- [12] 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*, pages 21057–21067, 2023. [1](#)
- [13] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4574–4584, 2023. [1](#)