Supplementary Materials

1. Framework Details

1.1. Positional Encoding

Let the input be

$$\mu \in \mathbb{R}^3,$$

where μ is the input 3D Gaussian location.

Define L as the number of frequency bands, we have the following frequencies:

$$\omega_i = 2^i, \quad i = 0, 1, \ldots, L-1.$$

For each frequency ω_i, we compute

$$\gamma_{\omega_i}(\mu) = \big[\sin(\omega_i \mu),\ \cos(\omega_i \mu)\big].$$

Then, we stack across all L frequency bands (original input included):

$$\gamma(\mu) = \big[\mu, \gamma_{\omega_0}(\mu), \gamma_{\omega_1}(\mu), \ldots, \gamma_{\omega_{L-1}}(\mu)\big].$$

The output dimension is

$$D_{pos} = 3 \cdot \Big(1 + 2L\Big).$$

We use 12 frequency bands, therefore the dimension after positional encoding is 75.

1.2. Mapping Network

The mapping network consists of three hidden layers, each with 64 neurons, followed by an output layer with K neurons. All hidden layers use *ReLU* activation, while the output layer has linear activation.

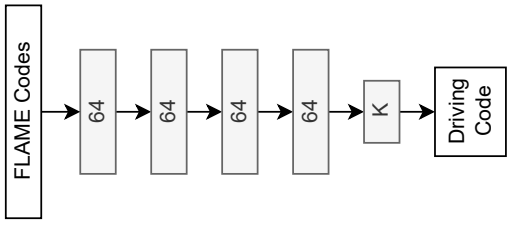

Figure 1. Architecture of the mapping network $\mathcal{M}$.

1.3. Decoder Network

Our decoder $\mathcal{D}$ consists of five sub-networks, each sharing the same architecture except for the input dimension, output dimension, and the activation function of the output layer. Each sub-network contains four hidden layers with 256 neurons each, followed by an output layer with D_{attr} neurons corresponding to the target Gaussian attribute (one of $\mu \in \mathbb{R}^3, \mathbf{s} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{R}^4, \alpha \in \mathbb{R}^1$, and $\mathbf{c} \in \mathbb{R}^3$). All hidden layers use *ReLU* activation, while the output layer employs different activation functions depending on the attribute. Table 1 summarizes the details.

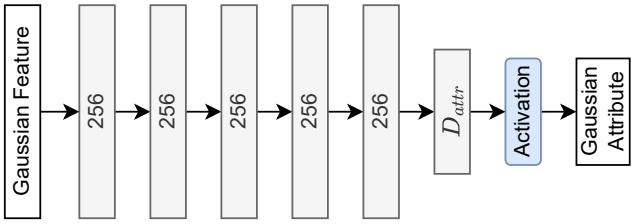

Figure 2. Architecture of the decoder network.

Network	D_{attr}	**Activation**
Position Offsets: $\Delta\mu$	3	scaled by 0.01
Rotations: $\mathbf{q}$	4	ℓ_2 normalization (quaternion)
Scales: $\mathbf{s}$	3	$\exp(\cdot) \times 0.0009$
Alphas: α	1	$\sigma(\cdot)$ (sigmoid)
SH Coefficients: $\mathbf{c}$	3	identity (no activation)

Table 1. **Summary of decoder networks.** Each head predicts a Gaussian attribute, followed by its respective activation.

2. Training

Training is performed in two stages. In stage one, the learning rates are set as follows: mapping network $\mathcal{D}$ (1e-5), multi-level latent field (5e-3), position offsets decoder (2e-4), scales decoder (1e-4), rotations decoder (1e-4), SH coefficients decoder (1e-4), and alphas decoder (1e-4). We train for 15,000 steps, with the learning rates reduced by half at step 10,000. In stage two, the learning rates are set as follows: multi-level latent field (5e-3), position offsets decoder (2e-4), scales decoder (1e-4), rotations decoder (1e-4), SH

Method	Subject	L1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
GaussianAvatars (CVPR'24) [2]	1	0.064238	24.311394	0.888906	0.132315
	2	0.044746	27.840605	0.918122	0.137628
	3	0.035400	25.876420	0.940552	0.066730
FlashAvatar (CVPR'24) [3]	1	0.060337	25.321769	0.889419	0.086785
	2	0.039521	30.044333	0.917574	0.082006
	3	0.033305	26.302053	0.937210	0.041791
RGBAvatar (CVPR'25) [1]	1	0.070550	25.914843	0.884510	0.110350
	2	0.060199	27.439313	0.889355	0.138269
	3	**0.028802**	**30.964819**	**0.947895**	0.051413
Gaussian Dejavu (WACV'25) [4]	1	0.057501	26.603426	0.906823	0.075991
	2	0.047117	29.381665	0.917544	0.080533
	3	0.030174	27.471601	0.945235	0.038314
ArchitectHead (Ours)	1	**0.051053**	**27.158812**	**0.914353**	**0.068233**
	2	**0.035130**	**30.705131**	**0.923910**	**0.071576**
	3	0.028992	27.999216	0.946682	**0.034507**

Table 2. **Quantitative comparisons of self-reenactment results on the PointAvatar [5] dataset.** Per-subject results are reported, with the best value of each evaluation metric in each column highlighted in bold.

		Subject								
Method	**Metric**	bala	biden	justin	malte_1	nf_01	nf_03	obama	person_0004	wojtek_1
GaussianAvatars (CVPR'24) [2]	L1 ↓	0.029	0.024	0.023	0.028	0.036	0.041	0.030	0.034	0.022
	PSNR ↑	25.977	28.684	27.591	27.709	26.575	25.759	26.929	24.746	28.840
	SSIM ↑	0.937	0.956	0.962	0.941	0.940	0.922	0.948	0.936	0.959
	LPIPS ↓	0.080	0.050	0.051	0.062	0.088	0.095	0.052	0.097	0.049
FlashAvatar (CVPR'24) [3]	L1 ↓	0.028	0.022	0.023	0.025	0.036	0.038	0.024	0.026	0.020
	PSNR ↑	28.303	29.751	27.509	28.891	26.682	26.676	28.785	28.097	30.561
	SSIM ↑	0.927	0.962	0.963	0.946	0.938	0.922	0.958	0.943	0.960
	LPIPS ↓	0.035	0.027	0.033	0.029	0.064	0.058	0.029	0.050	0.023
RGBAvatar (CVPR'25) [1]	L1 ↓	0.058	0.020	0.033	0.040	0.056	0.032	0.027	0.046	0.024
	PSNR ↑	21.603	**31.945**	25.397	27.354	24.503	29.310	30.047	27.779	30.310
	SSIM ↑	0.869	0.957	0.943	0.903	0.895	0.939	0.948	0.924	0.941
	LPIPS ↓	0.077	0.034	0.049	0.061	0.095	0.065	0.038	0.080	0.037
Gaussian Dejavu (WACV'25) [4]	L1 ↓	**0.021**	**0.020**	**0.019**	0.025	0.033	0.033	0.022	0.026	0.016
	PSNR ↑	30.451	30.997	29.038	28.975	27.836	28.740	30.472	28.178	32.814
	SSIM ↑	**0.956**	0.968	**0.970**	0.949	0.945	0.938	0.963	0.945	0.973
	LPIPS ↓	0.024	0.024	**0.030**	0.028	0.060	0.053	0.026	0.049	0.017
ArchitectHead (Ours)	L1 ↓	0.022	**0.020**	0.020	**0.022**	**0.032**	**0.028**	**0.021**	**0.023**	**0.015**
	PSNR ↑	**30.986**	31.195	**29.521**	**30.537**	**28.399**	**29.380**	**30.894**	**29.305**	**33.287**
	SSIM ↑	0.955	**0.970**	0.968	**0.956**	**0.950**	**0.949**	**0.966**	**0.950**	**0.976**
	LPIPS ↓	**0.023**	**0.022**	0.036	**0.024**	**0.059**	**0.045**	**0.025**	**0.044**	**0.015**

Table 3. **Quantitative comparisons of self-reenactment results on INSTA [6] dataset.** Per-subject results are reported, with the best value of each evaluation metric in each column highlighted in bold.

coefficients decoder (1e-4), and alphas decoder (1e-4). We train for 30,000 steps, with the learning rates halved at steps 20,000 and 25,000. Please refer to our code repository (see project homepage) for detailed implementation.

3. More Results

3.1. Per-Subject Self-Reenactment Results

We provide detailed evaluation results for each subject of the datasets in Tables 2 and 3.

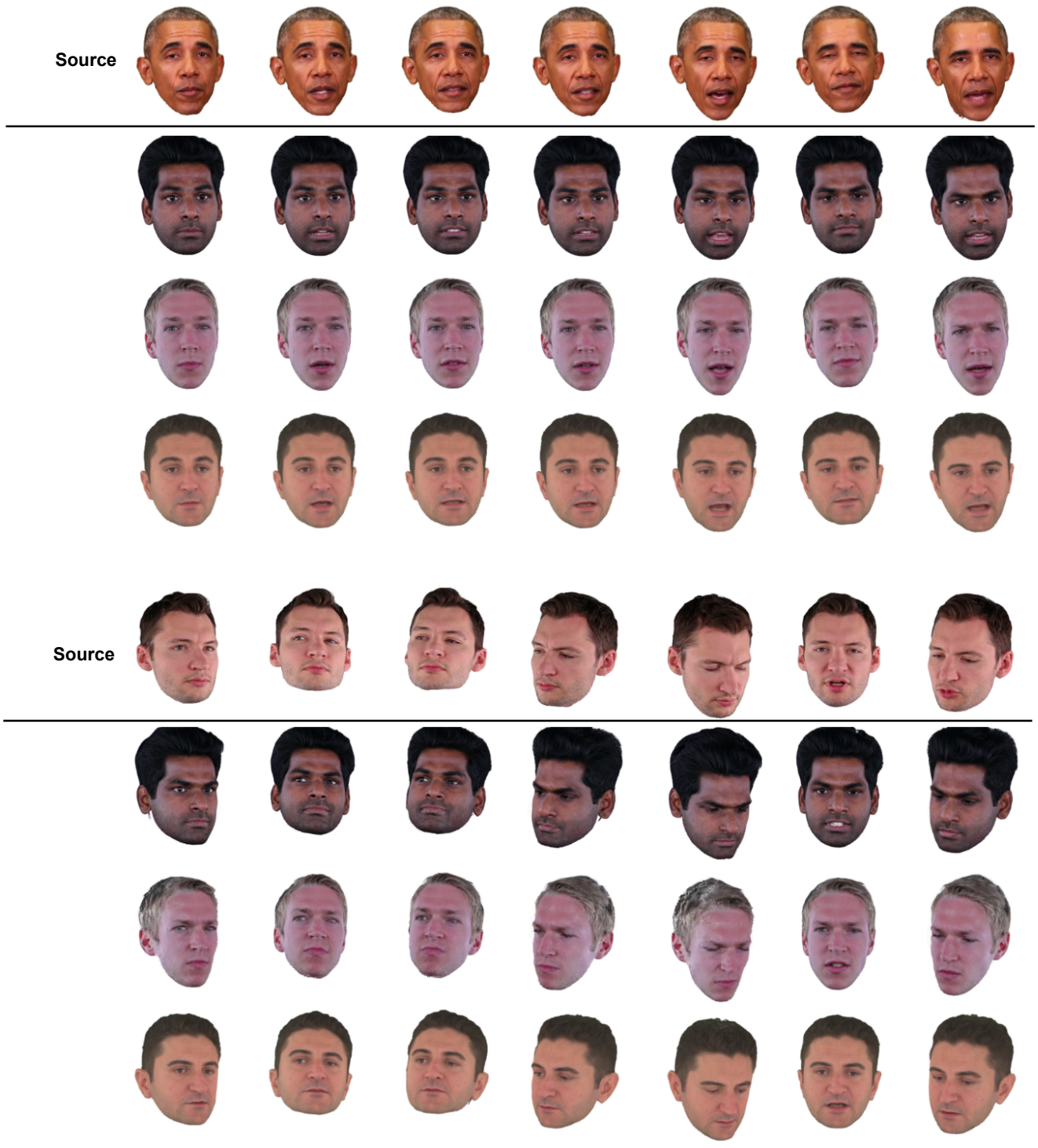

Figure 3. Cross-identity reenactment results.

3.2. More Cross-Reenactment Results

We provide more cross-reenactment results in Figure 3.

3.3. Varying Level of Details

We compare results across three LODs (0, 0.5, and 1.0) and different precisions (fp16 and fp32) during inference, while

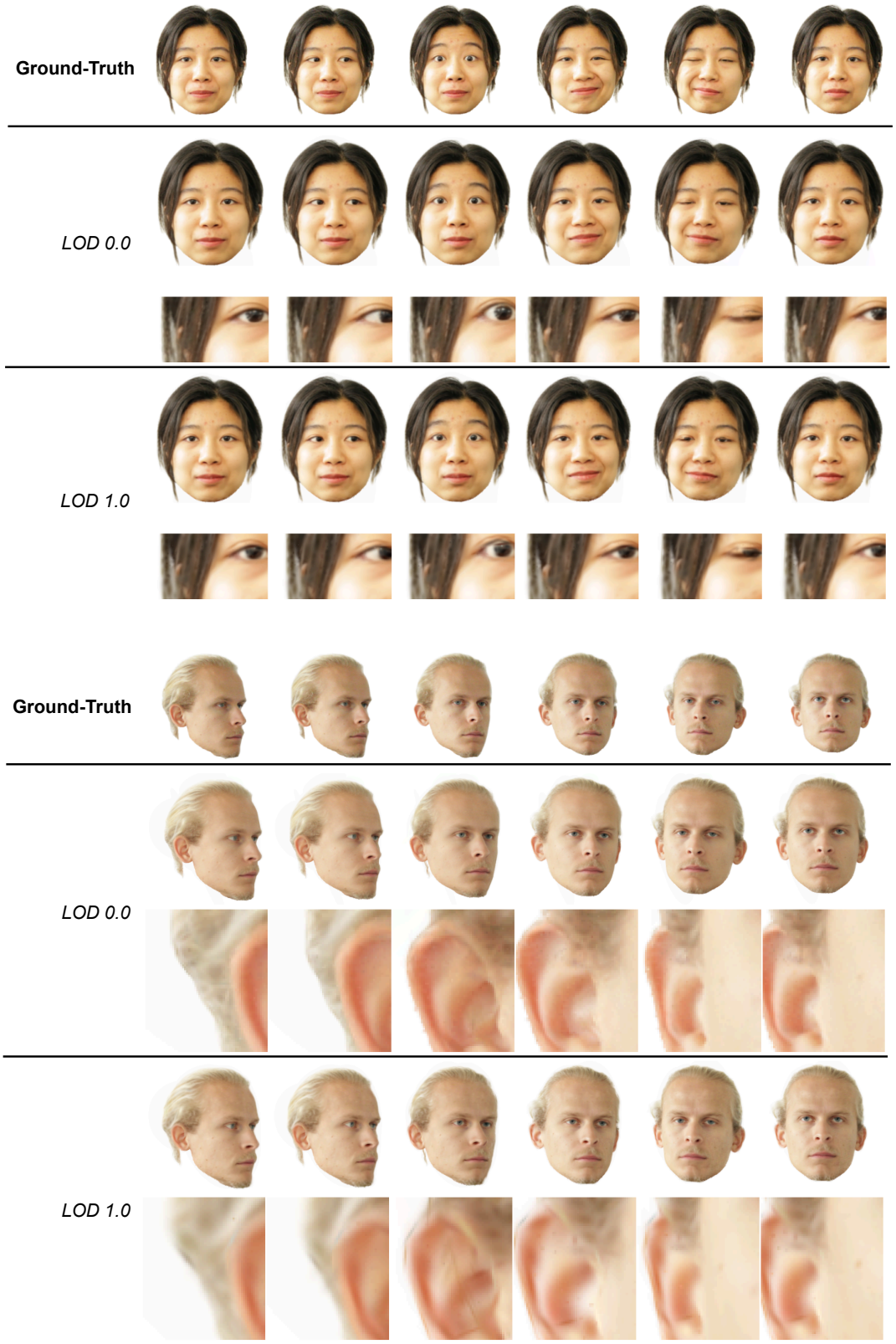

Figure 4. Self-reenactment results with LOD 0 and LOD 1.0.

training is performed in full precision. The results are summarized in Tables 4 and 5. The differences are negligible for most metrics, with only slight variation in PSNR, indicating that using the trained avatars in half precision does not noticeably degrade quality. Figure 4 presents self-reenactment results at LODs 0 and 1.0. When zoomed in, some artifacts and loss of details are visible in the LOD 1.0 results. Overall, the quality at the lowest LOD (1.0) remains reasonable.

Method (LOD)	L1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
ArchitectHead-fp16 (0.0)	0.038	28.615	0.928	0.058
ArchitectHead-fp16 (0.5)	0.039	28.508	0.927	0.061
ArchitectHead-fp16 (1.0)	0.041	28.337	0.922	0.072
ArchitectHead-fp32 (0.0)	0.038	28.621	0.928	0.058
ArchitectHead-fp32 (0.5)	0.039	28.514	0.927	0.061
ArchitectHead-fp32 (1.0)	0.041	28.342	0.922	0.072

Table 4. **ArchitectHead** on **PointAvatar** dataset, averaged over 3 subjects. Best values of different precisions under the same LOD level are highlighted.

Method (LOD)	L1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓
ArchitectHead-fp16 (0.0)	0.022	30.379	0.960	0.032
ArchitectHead-fp16 (0.5)	0.023	30.303	0.957	0.035
ArchitectHead-fp16 (1.0)	0.025	30.131	0.951	0.043
ArchitectHead-fp32 (0.0)	0.022	30.389	0.960	0.032
ArchitectHead-fp32 (0.5)	0.023	30.313	0.957	0.035
ArchitectHead-fp32 (1.0)	0.025	30.139	0.951	0.043

Table 5. **ArchitectHead** on **INSTA** dataset, averaged over 9 subjects. Best values of different precisions under the same LOD level are highlighted.

References

[1] 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*, pages 10747–10757, 2025. 2

[2] 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

[3] Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. Flashavatar: High-fidelity digital avatar rendering at 300fps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[4] Peizhi Yan, Rabab Ward, Qiang Tang, and Shan Du. Gaussian déjà-vu: Creating controllable 3d gaussian head-avatars with enhanced generalization and personalization abilities. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 276–286. IEEE, 2025. 2

[5] 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. 2

[6] 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. 2