Relightable Gaussian Codec Avatars –Supplemental Document–

Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, Giljoo Nam

Codec Avatars Lab, Meta

A. Network Architecture

Our head decoder consists of a view-independent decoder and a view-dependent decoder. An expression latent code $\mathbf{z} \in \mathbb{R}^{256}$ is first fed into a single linear layer with a leaky-ReLU, and then reshaped into $256 \times 8 \times 8$. Similarly, the gaze direction of each eye is fed into a linear layer with a leaky-ReLU, and then reshaped into $16 \times 2 \times 2$ for each. The gaze features are then only concatenated where the eye balls are located in the UV space, with the rest zero-padded. For view-dependent decoding, we take the unit vector direction from the rendering camera to the head center, and feed it into a linear layer with a leaky-ReLU to obtain a 8-dim latent feature, which is repeated across spatial dimensions for view-conditioning. The input features are concatenated and then fed into both decoders. Both the view-independent and view-dependent decoders consist of multiple up-sampling layers based on a transpose convolutional layer (4×4 kernel, stride 2) followed by a leaky-ReLU with channel sizes of $(272, 256, 128, 128, 64, 32, 16, 125)$ and $(280, 256, 128, 128, 64, 32, 16, 4)$ respectively. The eye decoder also uses a similar design while an input spatial resolution to the up-sampling layers of 4×4. The relative head rotation and position are simply repeated across the spatial dimensions. We also concatenate a visibility mask of eyeballs in UV space by jointly rasterizing the coarse head mesh and the eyeballs to account for the shadows cast by the eyelids. The channel sizes of both view-independent and view-independent layers are $(23, 256, 128, 128, 64, 64, 122)$, $(31, 256, 128, 128, 64, 64, 7)$ respectively. Note that we use weight normalization [7] for all linear layers and up-sampling layers, and untied bias [5, 6] for all up-sampling layers.

B. Discussion: Appearance Representation

In this section, we describe how we derive our specular term from the following rendering equation [3]:

$$\mathbf{c}(\boldsymbol{\omega}_o) = \int_{\mathbb{S}^2} \mathbf{L}(\boldsymbol{\omega}_i) V(\boldsymbol{\omega}_i) \rho(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \max(0, \boldsymbol{\omega}_i \cdot \mathbf{n}) \mathrm{d}\boldsymbol{\omega}_i, \tag{S1}$$

where $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ are incoming and outgoing light directions, $\mathbf{L}$ is the incoming light intensity, V is the visibility term, ρ is the BRDF, and $\mathbf{n}$ is the surface normal. Assuming the specular BRDF is represented with the general microfacet model, the specular component of BRDF is defined as follows:

$$\rho_S(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) = \frac{F(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) S(\boldsymbol{\omega}_o) S(\boldsymbol{\omega}_i)}{\pi(\boldsymbol{\omega}_i \cdot \mathbf{n})(\boldsymbol{\omega}_o \cdot \mathbf{n})} D(\mathbf{h}) \tag{S2}$$

$$= M(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) D(\mathbf{h}), \tag{S3}$$

where F is the Fresnel term, S is the geometric attenuation term, and $\mathbf{h}$ is the halfway vector. Following Wang *et al*. [9], we parameterize the normal distribution function (NDF) $D(\mathbf{h})$ as spherical Gaussian $G_s(\mathbf{p}; \mathbf{q}, \sigma)$ (Eq. 6 in the main paper). According to Wang *et al*. [9], the remaining term M is smooth and can be approximated as a constant across each Gaussian. After a spherical warping (Eq. 17-22 in [9]), we approximate Eq. S3 as:

$$\rho_S(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \approx M(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) G_s(\boldsymbol{\omega}_i; \mathbf{q}, \sigma), \tag{S4}$$

where $\mathbf{q}$ is the reflection vector. By substituting Eq. S4 into Eq. S1, our specular term becomes:

$$\int_{\mathbb{S}^2} (V(\boldsymbol{\omega}_i) M(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \max(0, \boldsymbol{\omega}_i \cdot \mathbf{n})) \mathbf{L}(\boldsymbol{\omega}_i) G_s(\boldsymbol{\omega}_i; \mathbf{q}, \sigma) \mathrm{d}\boldsymbol{\omega}_i. \tag{S5}$$

When $\sigma \ll 1$, the value inside the integral is 0 unless $\boldsymbol{\omega}_i$ is close to $\mathbf{q}$, which is determined by the input view $\boldsymbol{\omega}_o$. Therefore, we further approximate Eq. S5 by moving and combing all view-dependent terms together (denoted as v_k) except the incoming radiance $\mathbf{L}$ and NDF G_s as follows:

$$\mathbf{c}_k^{\text{specular}} = v_k(\boldsymbol{\omega}_o) \int_{\mathbb{S}^2} \mathbf{L}(\boldsymbol{\omega}_i) G_s(\boldsymbol{\omega}_i; \mathbf{q}, \sigma) \mathrm{d}\boldsymbol{\omega}_i. \tag{S6}$$

Importantly, we parameterize $v_k(\boldsymbol{\omega}_o)$ using a neural network, enabling end-to-end optimization with the remaining components to faithfully reproduce image observations. Thus, our model is flexible enough to represent specular reflection beyond the general microfacet model [9] or single-bounce reflection. We empirically find that this simple formulation is fast to compute, and stable to optimize. It also

Table S1. **Ablation Study**. The top three techniques are highlighted in red, orange, and yellow, respectively. We use 3D Gaussians with the explicit eye models for the geometric representations.

Method	Metrics		
	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	34.042	0.858	0.148
Ours w/o monoSH	33.762	0.853	0.152
Ours w/o view-dep nml.	33.927	0.864	0.148
SG [9, 11]	33.778	0.855	0.147

supports modeling both diffuse and highly reflective areas in a unified manner. In our paper, we constrain the specular BRDF to monochrome to prevent the specular term from overfitting diffuse components. Supporting color changes in specular highlights caused by dielectric materials or multi-bounce specular reflection can be addressed in future work.

C. Ablation Study

In this section, we provide ablation studies to validate our key design choices.

Higher-order Monochrome SH. Our diffuse color is based on spherical harmonics. To support high-frequency shadows, our model decodes additional monochrome SH coefficients up to 8-th order. We compare our approach with one where we remove 4-th to 8-th order monochrome SH coefficients with the remaining components being identical. Fig. S1 shows that our approach captures more precise shadows. The quantitative evaluation in Tab. S1 also shows that adding the monochrome SH coefficients improves overall reconstruction accuracy. Note that while some recent works utilize explicitly computed shadow maps [1, 2, 8], this is intractable for real-time relighting with high-frequency environments. Improving the sharpness of shadows in real-time relighting even further is an interesting direction for future work.

View-dependent Normal. Another component in our appearance model is the view-conditioned surface normal. We compare our approach with one where we remove view-conditioning when decoding the surface normal. Interestingly, the improvement does not clearly appear in both qualitative and quantitative comparisons (see Tab. S1). We hypothesize that our view-conditioned visibility term can compensate for some of the errors caused by view-independent surface normals in cylindrical regions. While this allows the baseline using view-independent normals to achieve comparable performance under discrete point lights, this would likely cause inaccurate reflection on continuous environments. We keep our view-conditioned normals as this offers a more geometrically correct interpretation for the cylinder-like 3D Gaussians.

Spherical Gaussian Formulation. Prior works using spherical Gaussians [9, 11] typically use a different parametrization $G(\mathbf{p}; \mathbf{q}, \lambda, \mu) = \mu e^{\lambda(\mathbf{p}\cdot\mathbf{q}-1)}$. We compare our method with this formulation of spherical Gaussians with the remaining parts being identical. While the overall results are comparable quantitatively, Fig. S2 shows that our parameterization better captures sharp eye glints, which is critical for accurate all-frequency reflections.

Figure S1. **Ablation Study: Monochrome SH**. Compared to a held out frame (a), using higher-order monochrome SH coefficients (b) improves the sharpness of shadows compared to a model without them (c).

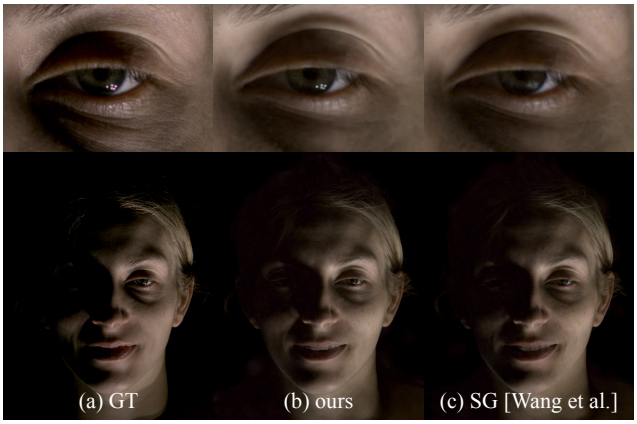
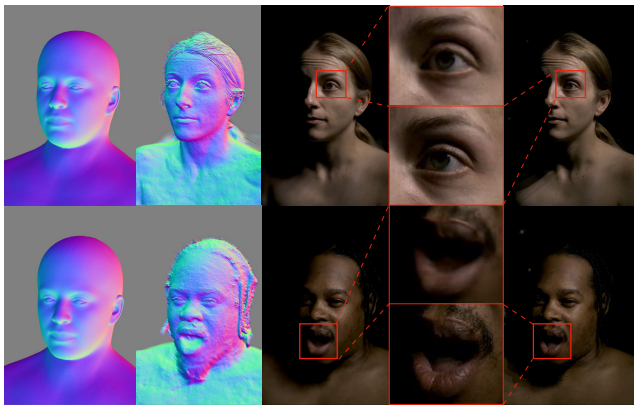


Figure S2. **Ablation Study: Spherical Gaussian Representation**. Compared to a held out frame (a), our angle-based SG formulation (b) leads to more accurate recovery of eye glints than the conventional cosine-based SG formulation [9] (c).

Person-specific mesh and non-rigid tracking required? We train our model with a generic head template as initialization regardless of facial expressions (Fig. S3 (a)). We also disable the geometry loss $\mathcal{L}_{\text{geo}}$ such that the positions of Gaussians are only updated through differentiable rendering. In other words, we use only the estimated rigid headpose and gaze directions as input. Although slightly

worse registration sometimes leads to lack of eye glints and blurrier extreme facial expressions, the model achieves surprisingly good reconstruction as shown in Fig. S3 (b). This indicates that our Gaussian-based representation is flexible enough to register even if the initialization is poor. The dependency on accurate non-rigid surface tracking can be optionally removed at the risk of slight quality degradation (e.g., lack of eye reflections).



(a) initialization (b) learned avatar (c) zoom-in (d) GT

Figure S3. **Ablation Study: Only Rigid Tracking**. We use a generic head template as the base mesh regardless of facial expressions (a). Compared to GT (d), our model with only rigid head pose tracking and a generic template achieves surprisingly good reconstruction (b, c).

Effect of the number of cameras. We train our decoder model with varying numbers of cameras to analyze the sensitivity of the method to capture setup specifics, and show results of novel view synthesis on a training frame (Fig. S4). Using as few as 32 cameras seems to yield good results, with 8 cameras showing noticeably degraded quality, and 16 cameras showing some artifacts, especially in the eyes. Conversely, using more than 32 cameras yields diminishing returns. We hypothesize that higher capacity modeling would be required to fully utilize the available data. (Note also that any rigid head motion present across the training frames creates additional virtual viewpoints—training on a single frame would yield much worse results).

Effect of the number of lighting conditions. We train our decoder model with varying numbers of light conditions and show an unseen light condition on a training frame (Fig. S5). We note two limitations of this study: (1) because we use temporal multiplexing, the comparisons use different numbers of training frames (as all frames from other light conditions need to be discarded), and (2) we cannot hold out physical lights as our light conditions trigger multiple lights simultaneously. However, the results show that using even 10% to 20% percent light conditions can yield acceptable results, potentially again limited by capacity and learning variance.

(a) 8 cameras (b) 16 Cameras (c) 32 Cameras (d) 149 Cameras

Figure S4. **Ablation Study: Number of cameras for decoder training**. We vary the number of cameras used for rendering supervision (a) 8 cameras, (b) 16 cameras, (c) 32 cameras, (d) the full 149 cameras. We show results of novel view generation on a training frame.

(a) 10+1 Light Conditions (b) 30+1 (c) 120+1 (d) 360+1 (e) 577+1 (Training Sample) (f) GT Image

Figure S5. **Ablation Study: Number of light conditions used in training.**. We vary the number of light conditions used for rendering supervision (a) 10+1 (10 partial illuminations and 1 uniform illumination), (b) 30+1, (c) 120+1, (d) 360+1, (e) the full set of illuminations (including the test sample), and (f) the ground truth image. We show results on held out illuminations for a training frame and camera.

D. Performance

For all identities, we use $1024 \times 1024 = 1$ Mi Gaussians for the evaluation and results on the paper, and $512 \times 512 = 256$ Ki Gaussians for the VR demo shown in the video. We observe that increasing the number of Gaussians leads to quality improvement at the cost of slower decoding and rendering. The 1024^2 model takes 12.84 ms for splatting, and the 512^2 model takes 6.40 ms for splatting on NVIDIA A100. We use 512^2 for the VR demo to improve the framerate. We do not apply any pruning of Gaussians. Tab. S2 shows the inference time of each method. All Gaussian-based models including ours converge within 3 days and MVP-based models require twice as many 032 iterations (400 K) for convergence.

Table S2. **Performance of each method**.

	Geometry	Appearance	Inference (ms)
A	Ours w/ EEM	EyeNeRF [4]	35
B		Ours	31
C	Ours	EyeNeRF [4]	20
D		Linear [10]	6
E		Ours	18
F	MVP [6]	EyeNeRF [4]	43
G		Linear [10]	6
H		Ours	34

E. Ethical Concerns

Our model is only applied to a few consenting subjects captured in a dense multiview capture system. In addition, the expression latent space is personalized for each individual to capture subtle expressions. These effectively limit the use case to driving ones' own avatars only with their consent.

References

[1] Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. Deep relightable appearance models for animatable faces. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021. 2

[2] Shun Iwase, Shunsuke Saito, Tomas Simon, Stephen Lombardi, Timur Bagautdinov, Rohan Joshi, Fabian Prada, Takaaki Shiratori, Yaser Sheikh, and Jason Saragih. Relightablehands: Efficient neural relighting of articulated hand models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16663–16673, 2023. 2

[3] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 1

[4] Gengyan Li, Abhimitra Meka, Franziska Mueller, Marcel C Buehler, Otmar Hilliges, and Thabo Beeler. Eyenerf: a hybrid representation for photorealistic synthesis, animation and relighting of human eyes. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 4

[5] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018. 1

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

[7] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016. 1

[8] Kripasindhu Sarkar, Marcel C. Buehler, Gengyan Li, Daoye Wang, Delio Vicini, Jérémy Riviere, Yinda Zhang, Sergio Orts-Escolano, Paulo Gotardo, Thabo Beeler, and Abhimitra Meka. Litnerf: Intrinsic radiance decomposition for high-quality view synthesis and relighting of faces. In *ACM SIGGRAPH Asia 2023*, 2023. 2

[9] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10. 2009. 1, 2

[10] 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 Proceedings*, 2023. 4

[11] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 2