Supplementary Material for PhysHead: Simulation-Ready Gaussian Head Avatars

Berna Kabadayi[1,3] Vanessa Sklyarova[1,2] Wojciech Zielonka[4] Justus Thies[1,4] Gerard Pons-Moll[3,5,6]

[1]Max Planck Institute for Intelligent Systems [2]ETH Zürich [3]University of Tübingen
[4]Technical University of Darmstadt [5]Tübingen AI Center [6]Max Planck Institute for Informatics

This supplementary material provides additional details and results for PhysHead.

1. Evaluation of Hair Dynamics and User Study

We conducted a preliminary user study with 26 participants which evaluated *physical plausibility* and *structural coherence*, finding our method significantly more plausible and coherent (Tab. 1). In addition, we provide quantitative metrics that capture the non-rigid behavior of dynamic hair using *E. Variance PC1 (%)* (lower indicates dynamic hair) and *Temporal Smoothness* in PCA temporal space. In Fig. 1, we also show a graph of the std. dev. of the hair geometry in PCA space which effectively factors out rigid transformations and gives an indication of the hair deformations. Especially, the std. dev. of the third principle component shows the hair deformation during the nodding sequence. GA [8] and GHA [12]) have almost static hair (std. dev. does not change), while ours is dynamic.

Metric	GA[8]	GHA[12]	Ours
Explained Variance PC1 (%) ↓	95.42	96.16	**93.28**
Temporal Smoothness (TS) ↓	6.23	2.13	**0.863**
User Study – Physical Plausibility ↑	0	1.5 %	**98.5 %**
User Study – Structural Coherence ↑	8.5 %	3.1 %	**88.5 %**

Table 1. E. Variance PC1 (%) shows how much of the motion of hair 3D Gaussians is captured by a single principle component. TS measures frame to frame change in PCA space.

2. Implementation Details

2.1. Optimization

Following Gaussian Avatars [8], we train the dynamic face region for 600k iterations using the Adam optimizer [4], adopting their hyperparameter configuration. The hair region is optimized separately for 50k iterations. We first optimize colors in the visible regions to obtain stable estimates, then introduce a color consistency loss to propagate these colors to the invisible regions after 3k iterations.

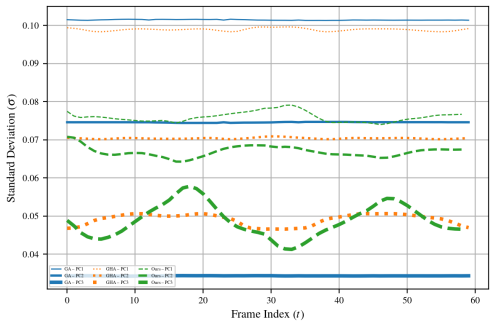

Figure 1. Analysis of the riggidity of the hair geometry on a nodding sequence. Our method (green) handles dynamic effects, others are static resulting in a constant std. dev. of geometry.

2.2. Dataset

We conduct our experiments on the Ava-256 dataset [7]. The list of actors used for our method is provided in Table 2.

Table 2. **Actors IDs.** List of actors from the Ava-256 [7] dataset used in our experiments.

Actors IDs.
20230313–0819–GCZ208
20230313–1653–RHL466
20230512–1009–ZEL435
20230810–1355–AJR151
20211006–0836–EID363
20230124–1415–KKF424
20230310–1106–FCT871
20230502–0816–ORZ494
20230602–1453–FMB793
20230308–1352–BDF920
20230726–1657–AYE877

2.3. Baselines

For comparison with GH [13], for each actor in Ava-256 [7], we split the cameras into train and test sets, resulting in 64 and 16 cameras, respectively, with every 5th camera used for testing. We use the expression *cheek002*. Gaussian Haircut [13] results are provided by the corresponding authors. We optimize the appearance of hairstyles obtained from Neural Haircut [10] on the training cameras and evaluate appearance results on the test set. Gaussian Avatars [8] and Gaussian Head Avatar [12] are run using the public repositories and trained on the same expressions and cameras.

2.4. Hair Simulation

Runtime of physics-based hair animation. Our hair simulation and sparse-to-dense interpolation run offline. Simulating 700 guide strands (16 points each) in Maya takes approximately 1s per frame on a MacBook Air (M3, 16GB RAM). The rendering of the Gaussian-based hair animation is real-time capable, taking roughly 0.0033s per frame, which is comparable to GA's [8] 0.0017s per frame for the same actor.

Hair Simulation Parameters. The physical parameters used for dynamic hair simulation are summarized in Table 3.

Table 3. **Hair system parameters used in our simulation.** The *Start Curve Attract* parameter varies slightly across different hairstyles.

Parameter	Value
Self Collide	1
Friction	0.51087
Stickiness	0.51087
Stretch Resistance	600
Compression Resistance	600.0
Bend Resistance	10
Twist Resistance	1.718
Extra Bend Links	1.0
Mass	2
Drag	0.65
Tangential Drag	0.096
Damp	0.25
Stretch Damp	1
Dynamics Weight	1
Static Cling	0.025
Start Curve Attract	2.0

From simulated guiding strands to dense hairstyle. Given a sparse set of hair strands, the motion data is represented as:

$$S_t \in \mathbb{R}^{N_s \times N_{\text{seg}} \times 3} \ , \tag{1}$$

where S_t represents the sparse strands at time t, N_s is the number of sparse strands, N_{seg} is the number of segments per strand, and each segment is represented in 3D space. In our experiments, we use $N_s = 700$ sparse strands, each consisting of 16 points (i.e., 15 segments) for the simulation.

For each dense strand d_i, we find the nearest $k = 10$ sparse strands based on their root positions (first segment at $t = 0$):

$$\mathcal{N}(d_i) = \text{KNN}(S_0[:, 0, :], d_i(0)) \ , \tag{2}$$

where $\text{KNN}(\cdot)$ finds the nearest neighbors based on Euclidean distance.

The influence of each sparse strand on a dense strand is computed using inverse distance weighting:

$$w_{i,j} = \frac{1}{d_{i,j} + \epsilon}, \quad w_{i,j} \leftarrow \frac{w_{i,j}}{\sum_{j=1}^{k} w_{i,j}} \ , \tag{3}$$

where $d_{i,j}$ is the Euclidean distance between the root of dense strand i and the root of its j-th nearest sparse neighbor, and ϵ is a small constant to avoid division by zero.

For each dense strand d_i at time t, the motion is computed using the weighted displacement of its nearest sparse strands:

$$\Delta S_t(i) = \sum_{j=1}^{k} w_{i,j} \cdot (S_t(j) - S_{t-1}(j)) \tag{4}$$

$$D_t(i) = D_{t-1}(i) + \Delta S_t(i) \tag{5}$$

where $\Delta S_t(i)$ is the weighted relative motion transferred from sparse to dense strands, and $D_t(i)$ is the updated dense strand position at frame t.

For hair reconstruction, we use a dense hairstyle consisting of 60,000 strands, each with 50 points, resulting in 49 Gaussian primitives per strand.

3. Preliminaries

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [2] reconstructs and renders static multi-view scenes from novel viewpoints. Kerbl et al. [2] model the scene using 3D Gaussians [5, 11], parameterized by a mean vector μ and a 3D covariance matrix $\mathbf{\Sigma}$:

$$G(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}. \tag{6}$$

To render these 3D Gaussians, they must be projected onto the 2D image plane [14]. However, perspective projection is inherently non-linear due to division by depth, making

direct computation of the 2D covariance challenging. To approximate this transformation, the first-order Taylor expansion is applied to compute the Jacobian matrix $\mathbf{J}$, which locally linearizes the mapping from 3D to 2D. The 2D covariance matrix is then obtained as:

$$\mathbf{\Sigma}' = \mathbf{J}\mathbf{\Sigma}\mathbf{J}^T, \tag{7}$$

where $\mathbf{\Sigma}'$ represents the projected 2D covariance matrix. The Jacobian matrix $\mathbf{J}$ is computed from the perspective projection function:

$$\mathbf{J} = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{fx}{z^2} \\ 0 & \frac{f}{z} & -\frac{fy}{z^2} \end{bmatrix}, \tag{8}$$

In this formulation, the Jacobian $\mathbf{J}$ accounts for the local changes in image-space coordinates to variations in 3D space. Since direct optimization of the covariance matrix $\mathbf{\Sigma}$ is difficult due to the requirement that it remains positive semidefinite, Kerbl et al. [2] instead decompose it into a scale matrix $\mathbf{S}$ and a rotation matrix $\mathbf{R}$, representing the 3D Gaussian as a 3D ellipsoid:

$$\mathbf{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T. \tag{9}$$

Furthermore, to approximate the diffuse component of the Bidirectional Reflectance Distribution Function (BRDF) [1], 3DGS employs spherical harmonics (SH) following the method of Ramamoorthi et al. [9]. Four bands of SH coefficients are used, leading to a 48-dimensional vector representation that captures view-dependent color and global illumination.

4. Additional Experiments

In this section, we provide additional comparisons of bald avatar appearance generation using our method and HairCUP [3] in Figure 2, along with extended results of our bald image generation pipeline in Figures 8, 9, and 10. Next, we compare hair dynamics and appearance with baseline methods in Figures 3 and 7. Finally, we present results of optimizing the head and hair as a single layer in Figure 5.

4.1. Generation of Bald Avatar

We use capabilities of Vision Language Model (VLM) to create realistic bald avatar. For our experiments we use the *gemini-2.5-flash-image-preview* model to process images of each actor captured from 16 different cameras at a single timestep. For each input image, our prompt is *make him/her bald, don't change skin color*. Overall, our API calls works reliably, though in rare cases (2 or less images for couple of actors), the generation fails due to security policies of the model. Despite these, the approach is effective for producing consistent bald-head images. After generating bald images, we overlay resize them to original input size and overlay with the original images. If face region is sharp, we consider that the generated view is consistent with the original capture, remove the background of generated [6]. Finally, we optimize the joint FLAME UV map for a single timestep using landmark and photometric losses with a

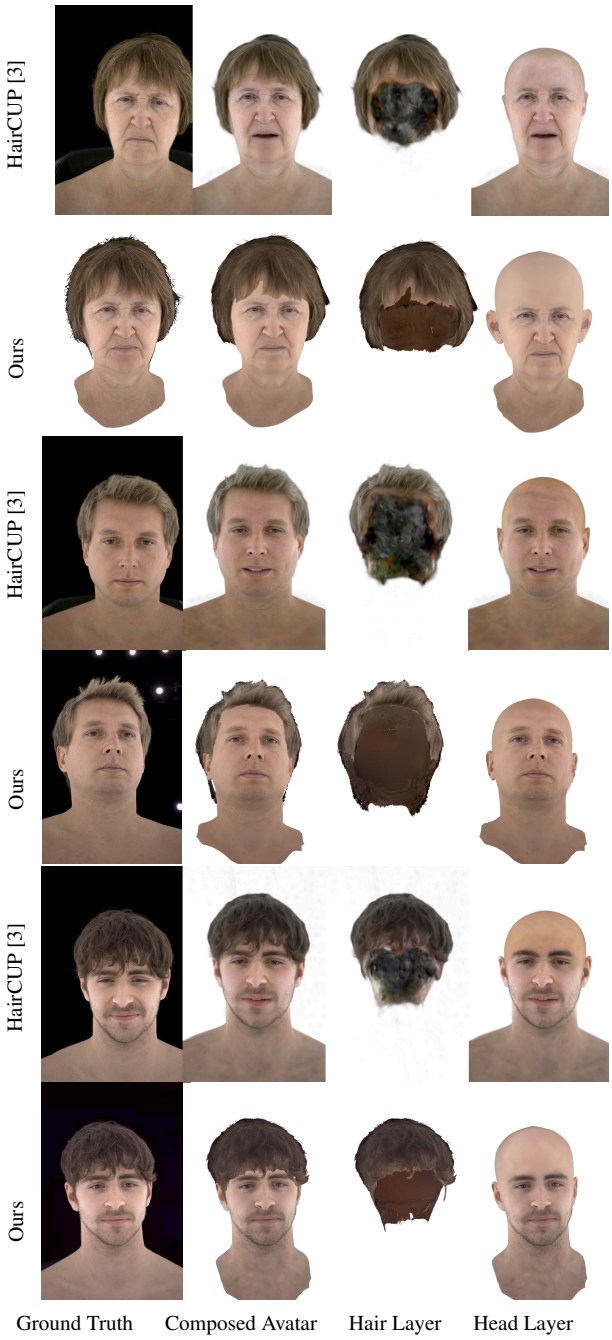

Figure 2. **Extended HairCUP [3] comparison.** HairCUP images are sourced from the original publication. As shown, HairCUP may produce artifacts and exhibits limited skin-tone variability across subjects, while our method mitigates these issues.

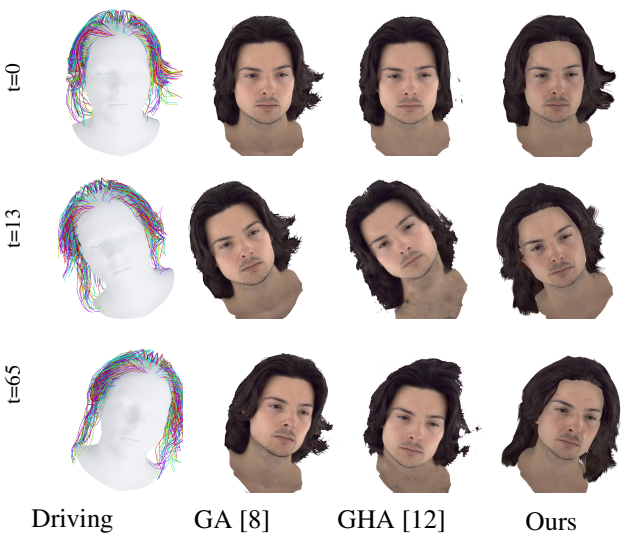

Figure 3. **Extended qualitative comparison.** All methods synthesize photorealistic avatars. GA produces rigid hair that remains nearly identical in each frame. GHA exhibits artifacts due to being controlled by a BFM-like model. In contrast, our method, PhysHead, benefits from a strand-based representation and generalizes well under novel driving signals.

differentiable rasterizer.

We provide additional comparisons of bald avatar generation using our method and HairCUP [3] in Figure 2. Note that HairCUP [3] code is not publicly available, so we use images provided in the paper. We noticed that HairCUP [3] exhibits limited skin-tone variability across different subjects, that might be an effect of using SDS loss. Our approach helps to better preserve color and produce consistent results. Furthermore, we show more results of our method in Figures 8, 9, and 10. We also include some failure cases of the VLM model in Figure 4. VLM sometimes alters the head pose, which breaks the multiview consistency required for further optimization. To mitigate this issue, we filter out such cases.

4.2. Qualitative Comparison with Baselines

First, we compare our learned hair appearance with Gaussian Haircut [13] in Figure 7, showing that our regularization loss significantly improves the realism of hair, particularly its internal structure. Finally, we present additional comparisons of dynamic hair appearance with Gaussian Avatars (GA) [8] and Gaussian Head Avatars (GHA) [12] in Figure 3. The main advantage of our physics-guided hair geometry is its ability to generalize to novel poses, in contrast to prior works that produce nearly static hair.

Figure 4. **VLM [13] failure cases.** Given the images in the first row, the VLM model generates bald images in the second row. However, the VLM model may alter the head pose (see the example in the third column). We filter out such cases and only use multiview-consistent views. Note that VLM-generated images are never used in our method; they are only employed to model regions such as the ear, hairline, or scalp.

4.3. Single Layer Representation

Jointly optimizing hair and head appearance in a single layer can produce artifacts during animation. Specifically, when animating strands with the physics engine, the skin may appear to peel along with the hair, as shown in Figure 5. To prevent this, we fully disentangle hair and head by optimizing the hair and head Gaussian layers separately, keeping the head Gaussian parameters fixed while optimizing hair.

Figure 5. **Single layer animation.** Single layer of optimization results in artifacts during animation despite the usage of hair strands.

4.4. Limitations

PhysHead operates on reconstructed strand geometry. In our experiments, we utilize NH [10], which fails to reconstruct curly hair, shown in Figure 6. However, our method is agnostic to the specific reconstruction method, benefits directly from future improvements.

Figure 6. **Curly Hair** Curly Hair results with GT and Ours.

Figure 7. **Extended Gaussian Haircut (GH) [13] comparison.** Gaussian Haircut captures the overall outer hair color nicely but exhibits artifacts in invisible regions. Our method propagates the outer color into inner strands, enabling consistent hair appearance.

Figure 8. **Additional VLM-Based Bald Image Generation Results.**

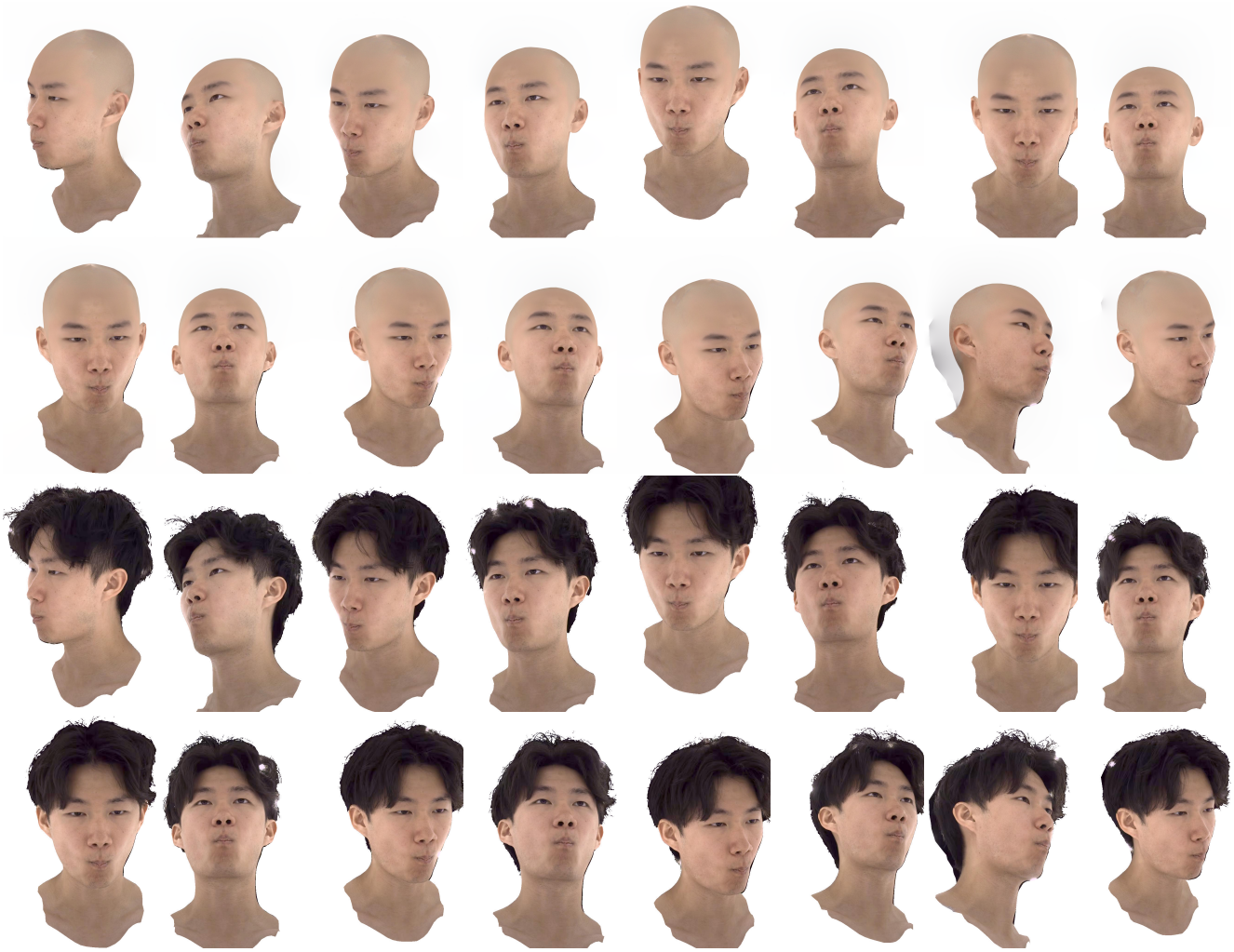

Figure 9. **Additional VLM-Based Bald Image Generation Results.**

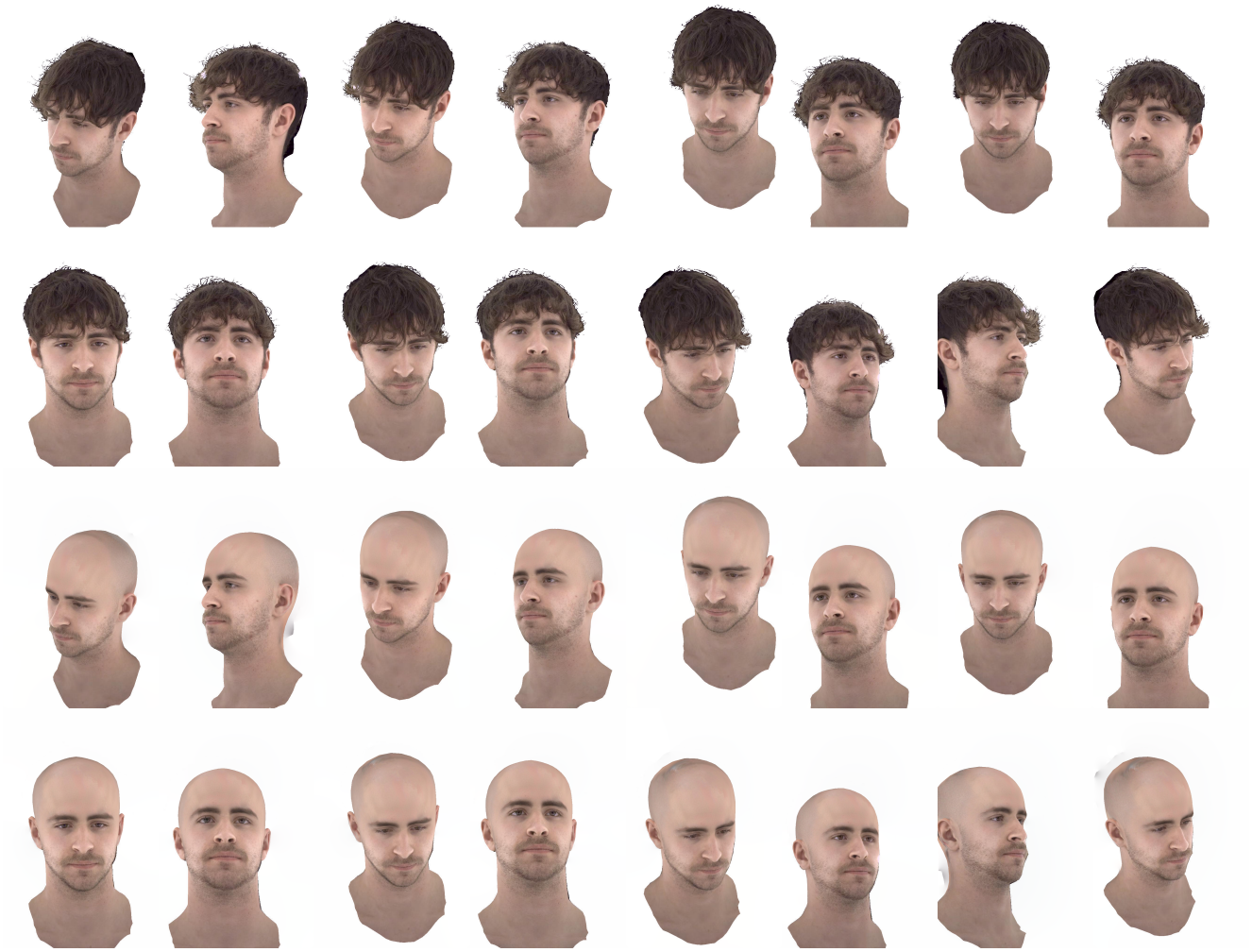

Figure 10. **Additional VLM-Based Bald Image Generation Results.**

References

[1] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *SIGGRAPH*, 1984.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42:1 – 14, 2023.

[3] Byungjun Kim, Shunsuke Saito, Giljoo Nam, Tomas Simon, Jason Saragih, Hanbyul Joo, and Junxuan Li. Haircup: Hair compositional universal prior for 3d gaussian avatars. *arXiv preprint arXiv:2507.19481*, 2025.

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[5] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Comput. Graph. Forum*, 40, 2021.

[6] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance, 2021.

[7] Julieta Martinez, Emily Kim, Javier Romero, Timur Bagautdinov, Shunsuke Saito, Shoou-I Yu, Stuart Anderson, Michael Zollhöfer, Te-Li Wang, Shaojie Bai, Chenghui Li, Shih-En Wei, Rohan Joshi, Wyatt Borsos, Tomas Simon, Jason Saragih, Paul Theodosis, Alexander Greene, Anjani Josyula, Silvio Mano Maeta, Andrew I. Jewett, Simon Venshtain, Christopher Heilman, Yueh-Tung Chen, Sidi Fu, Mohamed Ezzeldin A. Elshaer, Tingfang Du, Longhua Wu, Shen-Chi Chen, Kai Kang, Michael Wu, Youssef Emad, Steven Longay, Ashley Brewer, Hitesh Shah, James Booth, Taylor Koska, Kayla Haidle, Matt Andromalos, Joanna Hsu, Thomas Dauer, Peter Selednik, Tim Godisart, Scott Ardisson, Matthew Cipperly, Ben Humberston, Lon Farr, Bob Hansen, Peihong Guo, Dave Braun, Steven Krenn, He Wen, Lucas Evans, Natalia Fadeeva, Matthew Stewart, Gabriel Schwartz, Divam Gupta, Gyeongsik Moon, Kaiwen Guo, Yuan Dong, Yichen Xu, Takaaki Shiratori, Fabian Prada, Bernardo R. Pires, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, and Yaser Sheikh. Codec Avatar Studio: Paired Human Captures for Complete, Driveable, and Generalizable Avatars. *NeurIPS Track on Datasets and Benchmarks*, 2024.

[8] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. GaussianAvatars: Photorealistic head avatars with rigged 3D gaussians. In *CVPR*, pages 20299–20309, 2024.

[9] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. *SIGGRAPH*, 2001.

[10] Vanessa Sklyarova, Jenya Chelishev, Andreea Dogaru, Igor Medvedev, Victor Lempitsky, and Egor Zakharov. Neural Haircut: Prior-Guided Strand-Based Hair Reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[11] Yifan Wang, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM TOG*, 38:1 – 14, 2019.

[12] 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 (CVPR)*, 2024.

[13] Egor Zakharov, Vanessa Sklyarova, Michael J Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. Human hair reconstruction with strand-aligned 3d gaussians. *ArXiv*, 2024.

[14] Matthias Zwicker, Hans Rüdiger Pfister, Jeroen van Baar, and Markus H. Gross. Surface splatting. *SIGGRAPH*, pages 371–378, 2001.