

- Appendices -

In this document, we provide more details for the dataset, method, experiments, and more qualitative results, as an extension of Sec. 3 and Sec. 4 in the main paper. Please also refer to the video demo for dynamic hair results, comparison, ablation study, and more results.

A Dataset Details

As mentioned in the Sec. 1, due to the lack of real captures of dynamic hair deformations with accurate strand tracking, we create a new synthetic dataset capturing both hair geometry and appearance. The hair dataset breakdown is shown below: in Fig. 1, we present a variety of hairstyles in our synthetic hair dataset, including curly, wavy, blowout, ponytail and etc, and we show the distributions of hairstyle types and hair lengths.

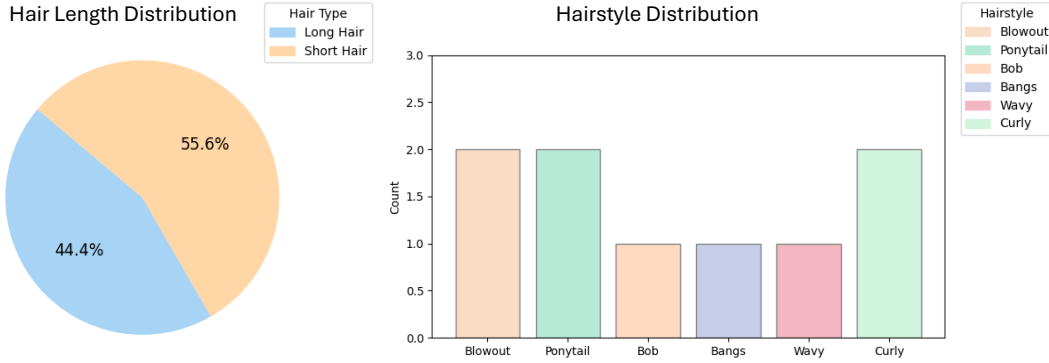


Figure 1. Hair length and hairstyle distribution

A.1 Geometry Dataset (Sec. 4)

To capture head motions, we rotate the head to record head movements. Instead of linear speed between two head positions, we apply spline-based interpolation to simulate varying motion speeds. This approach allows us to more effectively capture secondary hair motions when simulating the hair with head motions with realistic damping and inertial dynamics. For each hairstyle, we animated with XPBD-based physics simulation [7] driven by captured head motions, and for each hairstyle, we generate 100 motion sequences, and each motion sequence contains 100 frames. For each frame, we record the head mesh, deformed hair strands, and then post-process to get hair volume and pose volume. We show the hair geometry dataset generation pipeline in Fig. 2

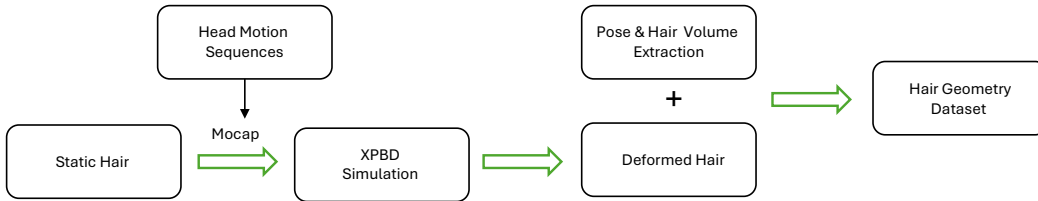


Figure 2. Hair Geometry Dataset Generation Pipeline

Fig. 3 presents samples from our hair deformation dataset, including static hair with strands, upper body SDF volume, static hair SDF volume, deformed hair strands, and deformed hair SDF volumes for each hair groom. For hair SDF volumes, hair point cloud (PC) is converted to an SDF grid by computing the distance from each voxel to the nearest point in the PC.

Training dataset. We train each hairstyle independently to obtain a hairstyle-specific hair deformation model. For each hairstyle, we use 90 motion subjects, resulting in a geometry training dataset of 9K frames. In the coarse stage, we randomly sample a frame and apply rigid transformations to the

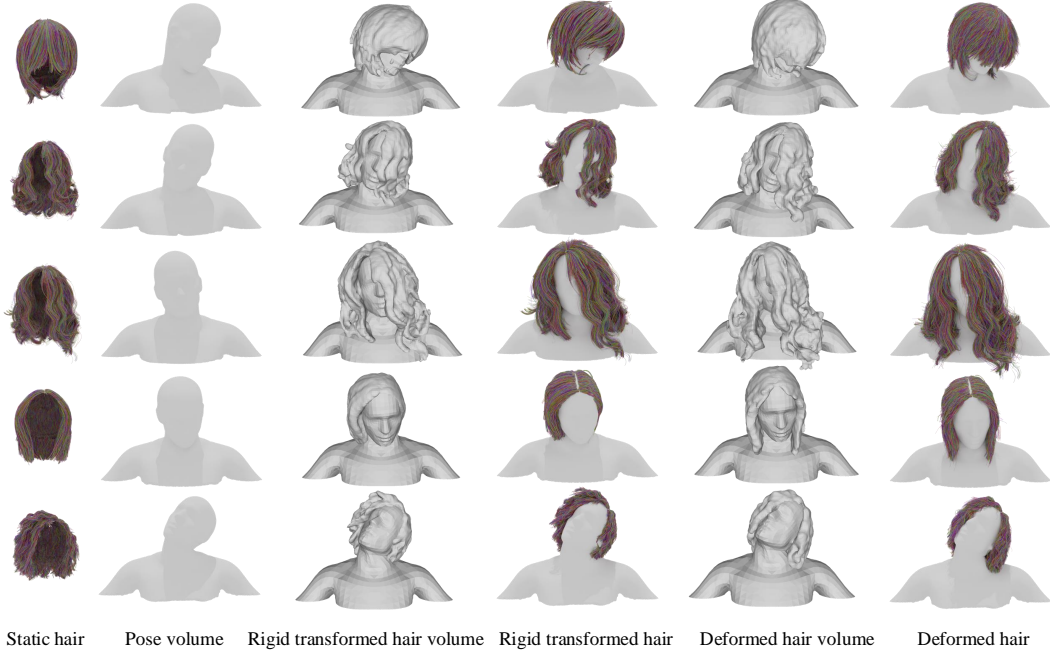


Figure 3. Training samples from the hair deformation dataset, including static hair, pose volume, rigidly transformed static hair, recorded hair volume, deformed hair, and corresponding recorded hair volume.

static hair using the sampled head pose. The network learns per-point displacements by iteratively sampling different poses. In the fine stage, we learn a flow vector field to deform the hair from frame $t-1$ to frame t , using randomly sampled points from the deformed hair at time $t-1$. Each training batch consists of 200K randomly sampled hair points.

Testing dataset. We evaluate our dynamic hair model across different hairstyles using 10 motion sequences, resulting in 1K frames. We report the average L_2 error between the predicted and ground-truth deformed hair, as well as the L_2 error of the estimated flow vectors (displacements between frames $t-1$ and t).

A.2 Appearance Dataset (Sec. 4)

Once obtaining the deformed hair sequences, we render multi-view videos of various hair grooms in Blender (particle system) under a consistent lighting setup to train our appearance model. Each frame has a resolution of 1024×1024 . For each hairstyle, we include diverse hair colors. Fig. 4 (left) shows the color variations for each hairstyle, (middle) illustrates the hair appearance dataset generation pipeline, and (right) presents a sample setup demonstrating how deformed hair is positioned for multi-view rendering. Each dynamic frame includes 24 views: 12 cameras evenly distributed horizontally around the object and 12 randomly positioned on the upper semi-sphere to capture diverse angles. Fig. 5 illustrates multi-view rendering samples for different hair grooms.

Training dataset We train our motion-dependent hair appearance model using multi-view video sequences. Each sequence contains 500 motion frames, and for each frame, we render images from 24 camera viewpoints distributed over a semi-sphere. This setup yields a total of 12K training images per hairstyle.

Testing dataset To evaluate our appearance model, we generate a testing dataset consisting of 100 frames featuring unseen hair motion. The test sequence is captured using a horizontal camera rotation arc, with each camera looking at the hair center. Viewpoints are sampled across a vertical range of 0° – 45° on the viewing semi-sphere, providing diverse angular coverage and challenging view-dependent appearance variations.

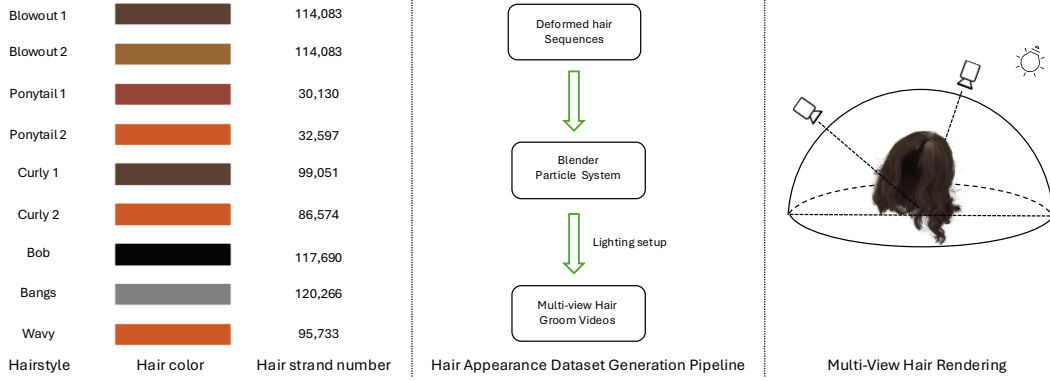


Figure 4. Hairstyle Color Variations and Hair Appearance Dataset Generation Pipeline

B Network and Evaluation Details

Hair, as a complex and dynamic structure, exhibits intricate spatial patterns and deformations that are challenging to model with lower-dimensional representations, the hair appearance also suffer challenges as the moving pattern caused occlusion. To address this, we propose two stage modeling: which Stage I and II as Hair Dynamics Model and Appearance Model. As summarized in Tab. 1, Stage I adopts a coarse-to-fine design: the coarse stage learns hair deformation from static hair, while the fine stage refines secondary dynamics (e.g., inertia and damping).

Module	Function
Stage I (coarse stage)	Learns hair deformation given head motions
Stage I (fine stage)	Adds secondary motion refinement
Stage II	Optimizes dynamic Gaussian hair appearance

Table 1. Overview of model stages and their corresponding functions.

For Hair Dynamics Model, we adopt a coarse-to-fine framework, representing static hair as a SDF volume to capture its complete 3D structure and spatial relationships. In this section, we present the architecture of our 3D CNN networks, $\mathcal{E}_{\text{pose}}$ and $\mathcal{E}_{\text{hair}}$, the implicit networks \mathcal{M} , \mathcal{D} , and \mathcal{D}^* , along with dataset and evaluation details.

B.1 Coarse-to-Fine Framework (Sec. 3.1)

In the **Coarse Stage**, we utilize a 3D U-Net architecture for $\mathcal{E}_{\text{pose}}$ and $\mathcal{E}_{\text{hair}}$ with input volumes of resolution $128 \times 128 \times 128$. We present the network structure in Fig. 6. The network employs separate encoders for static hair and pose, each comprising four convolutional blocks with filter sizes $F = 4, 2F, 4F$, and $8F$, utilizing 3D convolutions, LeakyReLU activations, and Instance Normalization to effectively extract hierarchical features.

The decoder mirrors the encoder with four deconvolutional blocks, progressively upsampling and reconstructing volumetric features while reducing the number of filters from $8F$ to F . Skip connections between corresponding encoder and decoder blocks preserve high-resolution details. With a base filter size of $F = 4$ and output channels of 16, the 3D U-Net effectively captures global context and fine-grained details of static hair, enabling precise learning of hair deformation by leveraging the static hair structure.

In the **Fine Stage**, the encoders $\mathcal{E}_{\text{pose}}$ and $\mathcal{E}_{\text{hair}}$ retain the same structure, with $\mathcal{E}_{\text{hair}}$ sharing weights across timesteps during training. The MLPs (\mathcal{M} , \mathcal{D}^*) consist of six fully connected layers with dimensions $[176, 512, 512, 256, 128, 3]$ and $[239, 512, 512, 256, 128, 3]$, respectively. Residual connections are employed to enhance feature representation by concatenating input features with intermediate outputs. Leaky ReLU is used as the activation function for all layers except the final one, ensuring non-linearity and stability. The final layer outputs a 3-dimensional vector, directly supervised by the ground truth displacement and 3D flow vector.



Figure 5. Training samples from the appearance dataset, which presents a static frame of various hair grooming and rendering results from six different camera views.

83 The fine stage estimates the flow vector from V_{hair}^{t-1} to V_{hair}^t . Since V_{hair}^t is not available, cross-
 84 attention [10] between V_{hair}^{t-1} and V_{hair}^{t-2} infers dynamics. “Ours w/o atten” uses only V_{hair}^{t-1} to estimate
 85 the flow vector. In the cross-attention, Q, K, V is V_{hair}^{t-2} , V_{hair}^{t-1} , V_{hair}^{t-1} . This approach leverages the
 86 immediate context from V_{hair}^{t-1} and longer-term dynamics from V_{hair}^{t-2} , enabling the model to capture
 87 motion patterns and ensure temporal consistency for a smooth and reliable estimation of V_{hair}^t .

88 B.2 Dynamic Hair Appearance Network and Implementation(Sec. 3.2)

89 For dynamic hair appearance optimization, the encoder $\mathcal{E}_{\text{hair}}$ encodes the deformed hair to cap-
 90 ture global motion features and 3D spatial information. A lightweight MLP \mathcal{D} with dimensions
 91 [169, 256, 256, 256, 128, 50] decodes the Spherical Harmonics coefficients for color, the scale factor
 92 along the hair length direction, and opacity. Positional encoding E for the position \mathbf{p} , tangent vector
 93 \mathbf{t} , and view direction \mathbf{d} follows the encoding function proposed in NeRF [8]. For Gaussian primitive
 94 initialization, we make a cylindrical Gaussian primitive attached to each hair segment with a length
 95 much greater than its radius. Similar to Gaussian Haircut [11], the scale of Gaussian primitives has
 96 only one degree of freedom, which is proportional to the length of the line segment, while the other
 97 two are fixed to a small predefined value.

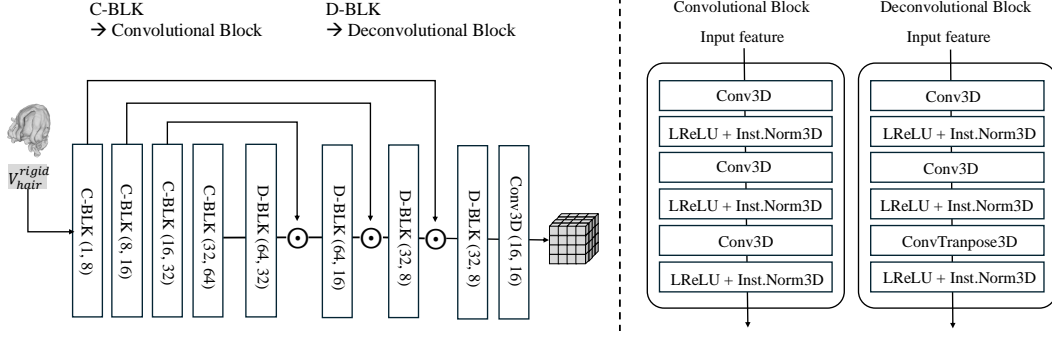


Figure 6. Left: Our 3D CNN networks for $\mathcal{E}_{\text{pose}}$ and $\mathcal{E}_{\text{hair}}$; Right: Convolutional and deconvolutional blocks.

B.3 Evaluation Details (Sec. 4)

Metrics. We evaluate hair motion over time by comparing the flow vectors between consecutive frames in the predicted and ground truth point cloud sequences. The flow vector for each point i at frame t in the predicted point cloud is defined as $\mathcal{F}_i^t = \mathbf{P}_i^{t+1} - \mathbf{P}_i^t$, where \mathbf{P}_i represents the predicted point cloud positions. Similarly, the flow vector in the ground truth point cloud is defined as $\hat{\mathcal{F}}_i^t = \hat{\mathbf{P}}_i^{t+1} - \hat{\mathbf{P}}_i^t$, where $\hat{\mathbf{P}}_i$ represents the GT point cloud positions. The total flow error at each timestep t over all points in the point cloud is given by $f_{\text{error}}^t = \frac{1}{N} \sum_{i=1}^N \|\mathcal{F}_i^t - \hat{\mathcal{F}}_i^t\|_2$, where N is the total number of points in the point cloud.

Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Ours w/o LPIPS & SSIM	27.82	0.8874	0.2100
Ours w/o LPIPS	28.01	0.8923	0.1987
Ours	28.12	0.9004	0.1881

Table 2. Ablation of different losses for appearance optimization.

Inference. We train and test our model on a single A100 GPU, with training times of approximately 20 hours for the dynamic hair coarse stage, 20 hours for the fine stage, and 26 hours for the appearance model. During inference, the dynamic hair model achieves 2.0 FPS for approximately 150K hair strands, significantly outperforming the XPBD [7] physics-based simulation engine that is used to generate the synthetic hair dataset, which runs at approximately 0.33 FPS. Meanwhile, dynamic hair novel view synthesis reaches 2.22 FPS, with faster speeds for fewer strands.

Runtime and memory performance analysis. DGH design is suitable for low-compute devices. Our implementation is an upper bound. Neural net inference can be accelerated with TensorRT [1] and additional strands linearly interpolated. For hair deformation runtime analysis, we report runtime with various strand numbers on RTX4090 GPU, compared with XPBD, as shown in Fig 7. Hair inference required 5GB, while high-quality grooms (150K+ strands) required 20GB. For dynamic hair rendering runtime analysis, our appearance model renders 1K images in 0.45 seconds (2.22 FPS), whereas Blender’s hair shader rendering takes 6 seconds per image (0.17 FPS).

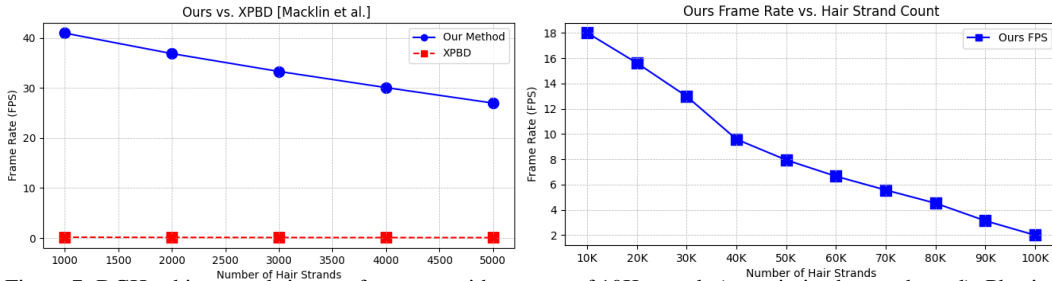


Figure 7. DGH achieves real-time performance with grooms of 10K strands (unoptimized upper bound). Physics-based simulation (XPBD) runs below 1 FPS with hair interpolation and equal high-quality photoreal rendering in Blender 3D engine.

119 **Appearance loss ablations.** We present the appearance loss ablation study in Tab. 2. For the final loss
 120 constraint, we incorporate both perceptual loss (LPIPS) and structural loss (SSIM), demonstrating
 121 that our full method achieves the best performance.

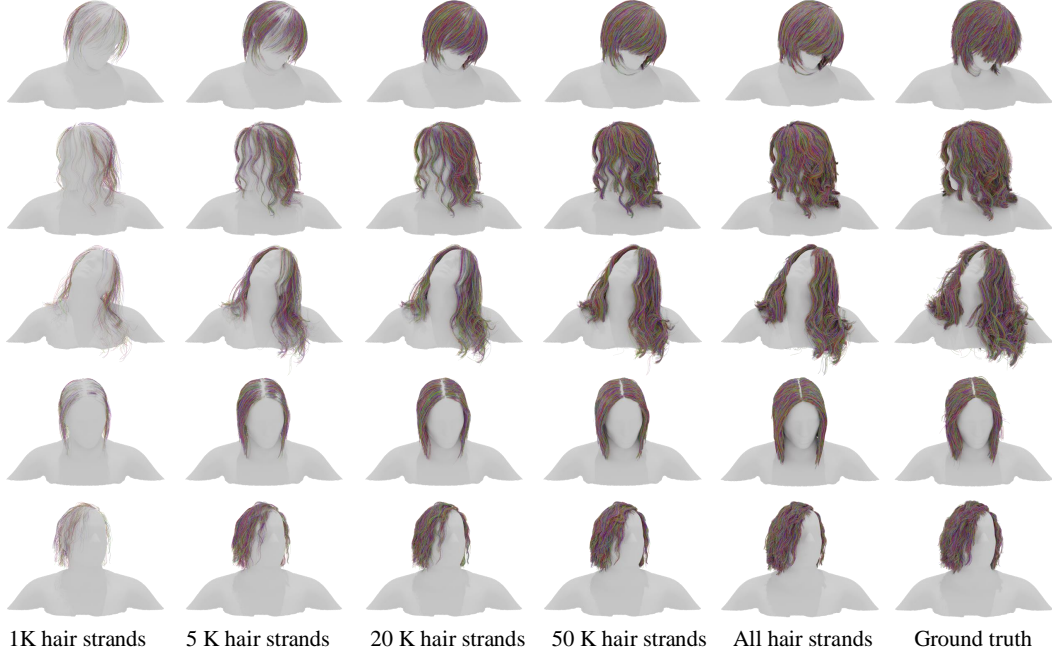


Figure 8. Hair deformation with varying strand counts. From left to right: increasing the inferred number of hair strands, totaling approximately 100K to 150K strands.

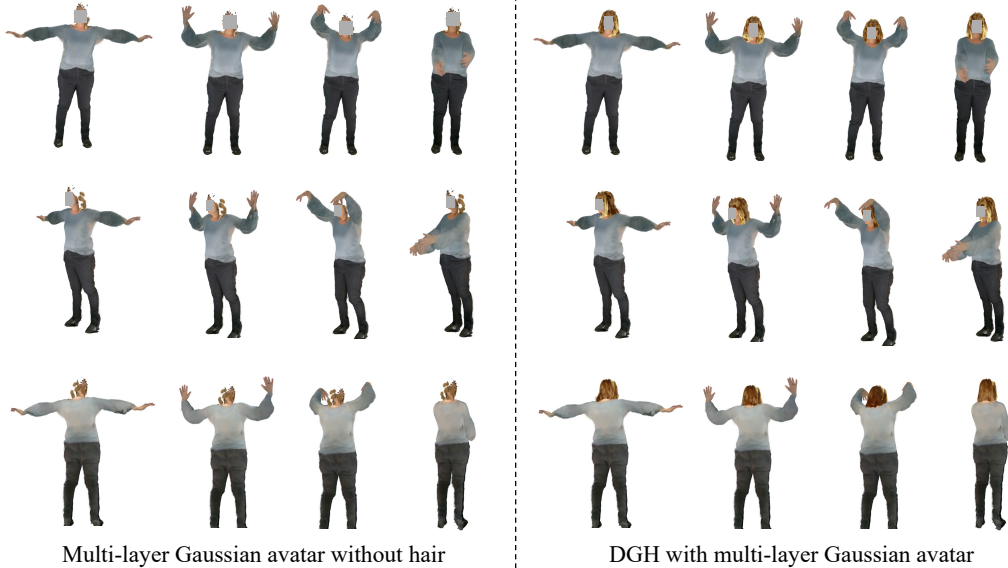


Figure 9. Our method merges the dynamic hair layer with a per-frame animatable Gaussian avatar. By employing a multi-layer Gaussian representation (left), we obtain an avatar without the hair layer. Applying DGH within this multi-layer framework enables realistic avatar re-animation (right). *Note: The human face is masked to comply with privacy regulations.*

C Qualitative Results

C.1 Hair Deformation.

In Fig. 8, we present the hair deformation test on arbitrary head poses and various hairstyles. We also present hair deformation results tested with varying numbers of hair strands, ranging from 1K to the full set of approximately 150K strands. Our volumetric implicit function robustly infers hair deformation for arbitrary strand numbers without requiring post-processing or hair interpolation.

C.2 Dynamic Hair Novel-View Synthesis.

Baseline settings. We compare our DGH model with two baseline methods: 3DGS [5] and Gaussian Haircut [11]. We train both baselines using their publicly released code on our canonical synthetic hair dataset, which includes 48 camera views covering the full hair. For 3DGS, we initialize Gaussian primitives by attaching them to the center of each hair segment. To maintain consistency with our dynamic sequences, we fix the number of Gaussians and disable density growth. We fix the opacity and positions of Gaussians, compute their orientation using hair tangents, and optimize three degrees of freedom of scale, color (represented using spherical harmonics). We use the Adam optimizer with a learning rate of 0.00025 for SH color features and 0.0005 for scale optimization. For Gaussian Haircut, we similarly optimize both color and scale, and the scale is constrained to be proportional to the length of each line segment to better capture hair geometry. Once each baseline model is trained on the static canonical hairstyle, we propagate the learned Gaussian parameters to dynamic sequences using our hair tracking model, which provides per-frame hair geometry.

Comparison and additional results. We present further results on dynamic hair novel-view synthesis in Fig. 10 and Fig. 11. To better evaluate appearance quality, we use ground-truth hair tracking and compare our appearance model against other Gaussian-based hair representation methods. Results are shown across multiple dynamic sequences and diverse hairstyles.

C.3 Multi-Layer Gaussian Avatars with Dynamic Hair.

Our dynamic hair Gaussian representation can be seamlessly integrated with pre-trained Gaussian avatar models for avatar re-animation and novel-view rendering, as shown in Fig. 12, 13, 14, 15 the render image resolution is 1024×1024 . Our appearance model supports only hair rendering, while the body is rendered using a separate pre-trained Gaussian model. For visualization, both hair and body Gaussians are rendered together. **Note that the static hair lacks gravity effects.** Using our hair tracking model, we represent hair strands as connected cylindrical Gaussians and achieve high-quality novel view synthesis through our motion-dependent appearance model. Unlike static hair, our dynamic hair model implicitly learns gravity effects from the dataset distribution, ensuring realistic rendering across dynamic hair motion sequences.

In Fig. 9, we further demonstrate how DGH can be seamlessly integrated with a multi-layer Gaussian avatar for realistic avatar re-animation. Since our DGH model primarily takes head motion as input, the resulting motion focuses on head-driven hair dynamics without incorporating body motion effects. Inspired by recent advancements in multi-layer Gaussian avatar modeling [6], realistic avatar editing and re-animation can be achieved by introducing additional Gaussian layers to represent various components such as hair and clothing. To merge an additional hair layer with the body avatar, our framework begins with a pre-trained head GS avatar re-animated using head motion inputs. The DGH module predicts dynamic Gaussian hair deformations directly from a half-body GS representation or a dense point cloud, which are then merged with the head GS to produce realistic, data-driven hair motion.

In contrast, physics-based methods re-animate a GS avatar by converting it into an explicit mesh, simulating hair motion with a physics engine, rendering it through a 3D pipeline, and re-converting the result into a 3DGS format before merging. Unlike such mesh-based pipelines that require multiple conversion and rendering stages, our DGH framework operates entirely in the Gaussian domain. By representing both hair and the upper body as volumetric Gaussians, it enables mesh-free deformation prediction, ensuring compatibility with Gaussian-based avatars and seamless integration into learning-based re-animation pipelines without rigging or simulation overhead.

172 D Limitation and Future Work

173 DGH currently handles hair-upper-body collisions. Environmental constraints and relighting of DGH
174 (via albedo inference) could be handled in future work. Due to the challenges of accurately tracking
175 real hair dynamics, precise deformation and per-timestep positions of hair strands are difficult to
176 record. Our model is currently trained and evaluated on synthetic data, focusing on generalizing
177 to head motions. In future work, we aim to create a large dynamic hair dataset based on static 3D
178 hair datasets [3, 4], leveraging a learned deformed hair prior. This identity-independent model could
179 robustly infer diverse hairstyles and perform well on real-world data.

180 In the second stage, a lightweight MLP combined with a differentiable rasterizer achieves high-fidelity
181 novel view rendering, balancing efficiency and quality. While fast for dense Gaussian primitives, the
182 current version is not real-time for dense hair (100K-150K hair strands). Future work could focus
183 on optimizing the MLP or leveraging caching methods [2, 9], such as viewpoint-aware or temporal
184 caching, to achieve real-time performance.

185 For multi-layer Gaussian avatars with dynamic hair applications, our model currently supports
186 integration with head avatars driven solely by head motion. Full-body effects such as jumping
187 and squirming under gravity are not yet modeled. In future work, we plan to incorporate explicit
188 material conditioning and full-body motion inputs to improve generalization across diverse hairstyles,
189 particularly with access to a more comprehensive 3D hairstyle dataset.

190 E Broader Impacts

191 As a positive impact, our work enhances the realism of Gaussian-based digital humans, enabling
192 more expressive virtual interactions in telepresence, AR/VR, and virtual production. As a negative
193 impact, increased photorealism may also raise ethical concerns, including potential misuse for identity
194 manipulation or deceptive content generation.

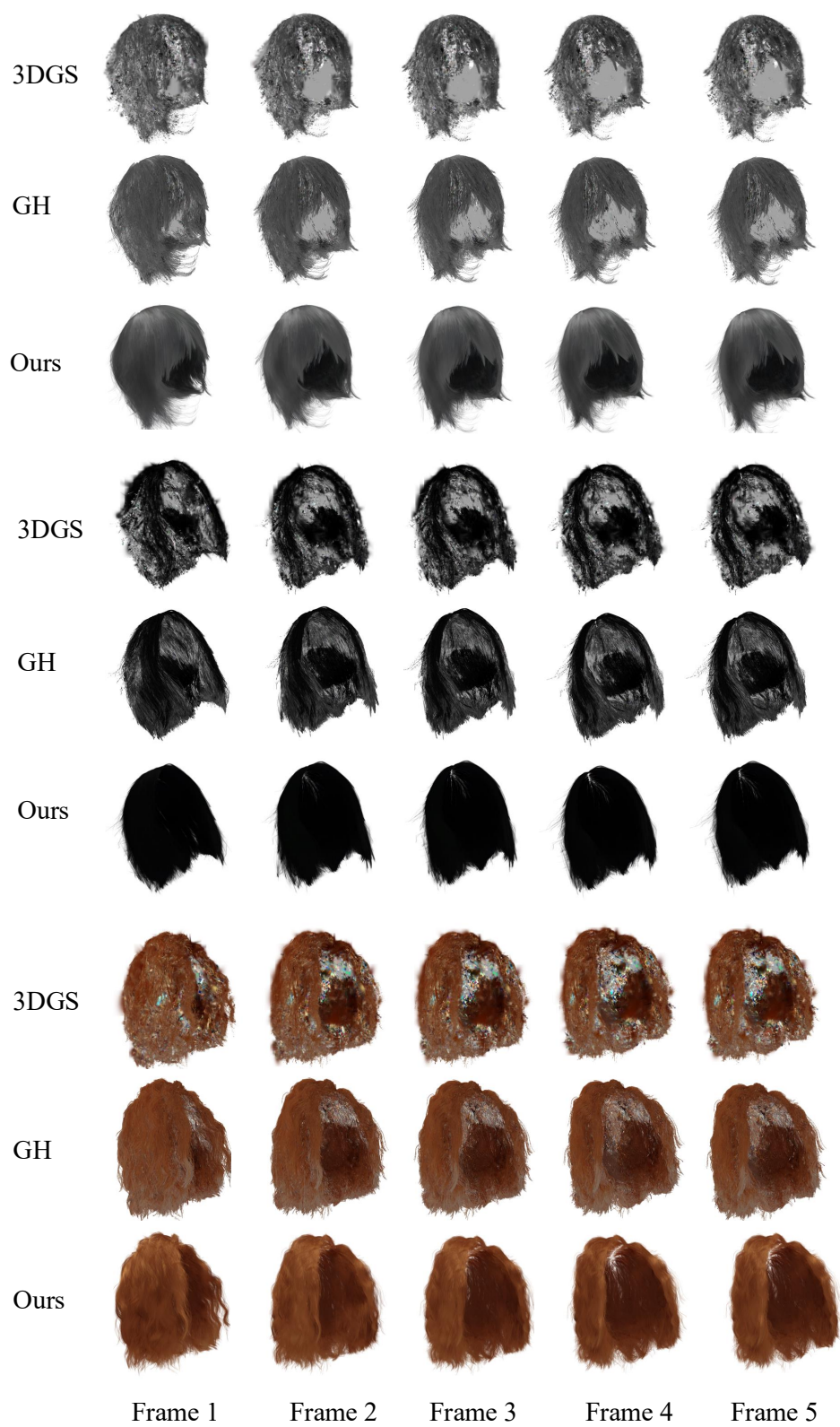


Figure 10. Comparison results for dynamic hair rendering

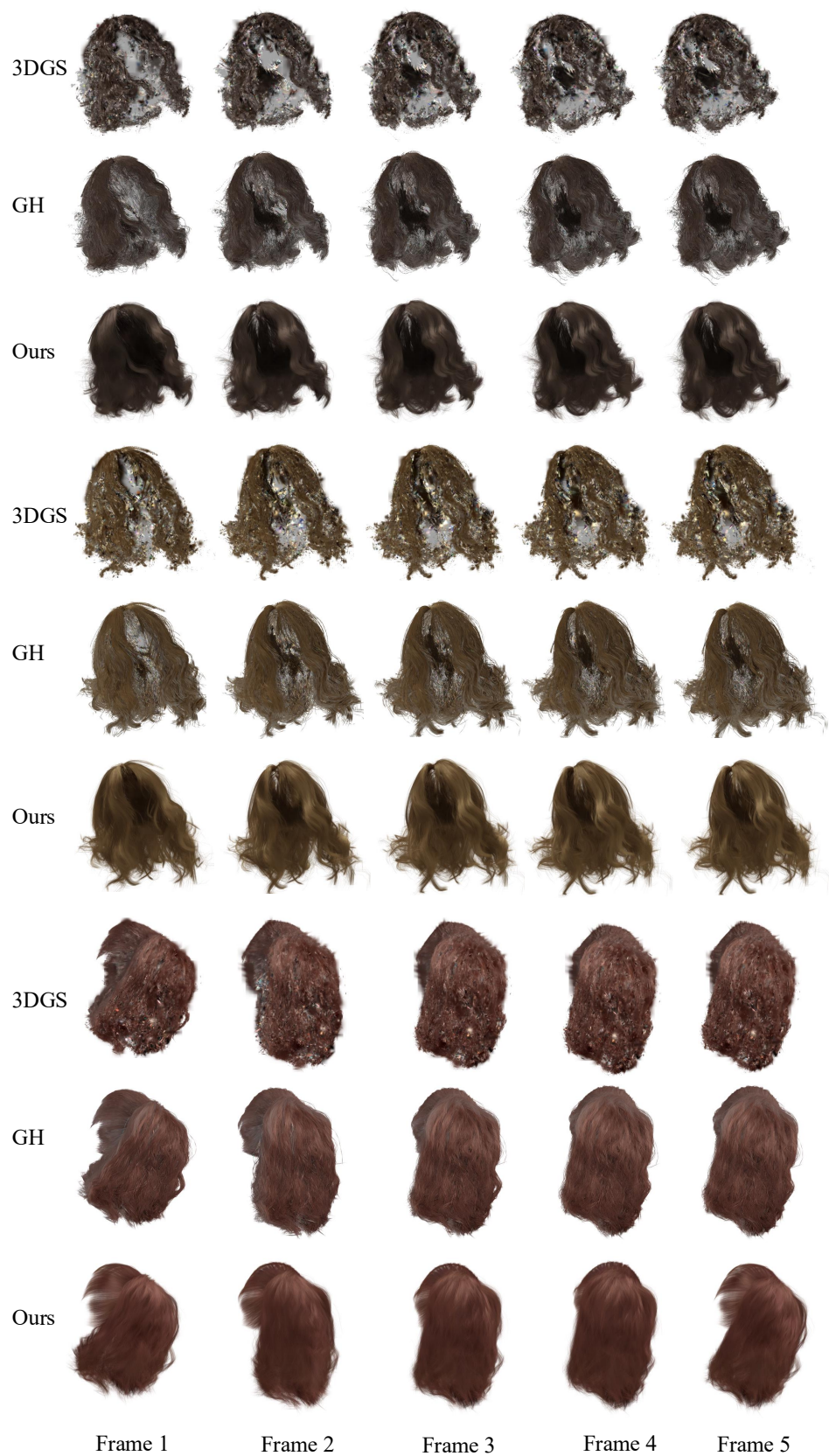


Figure 11. Comparison results for dynamic hair rendering

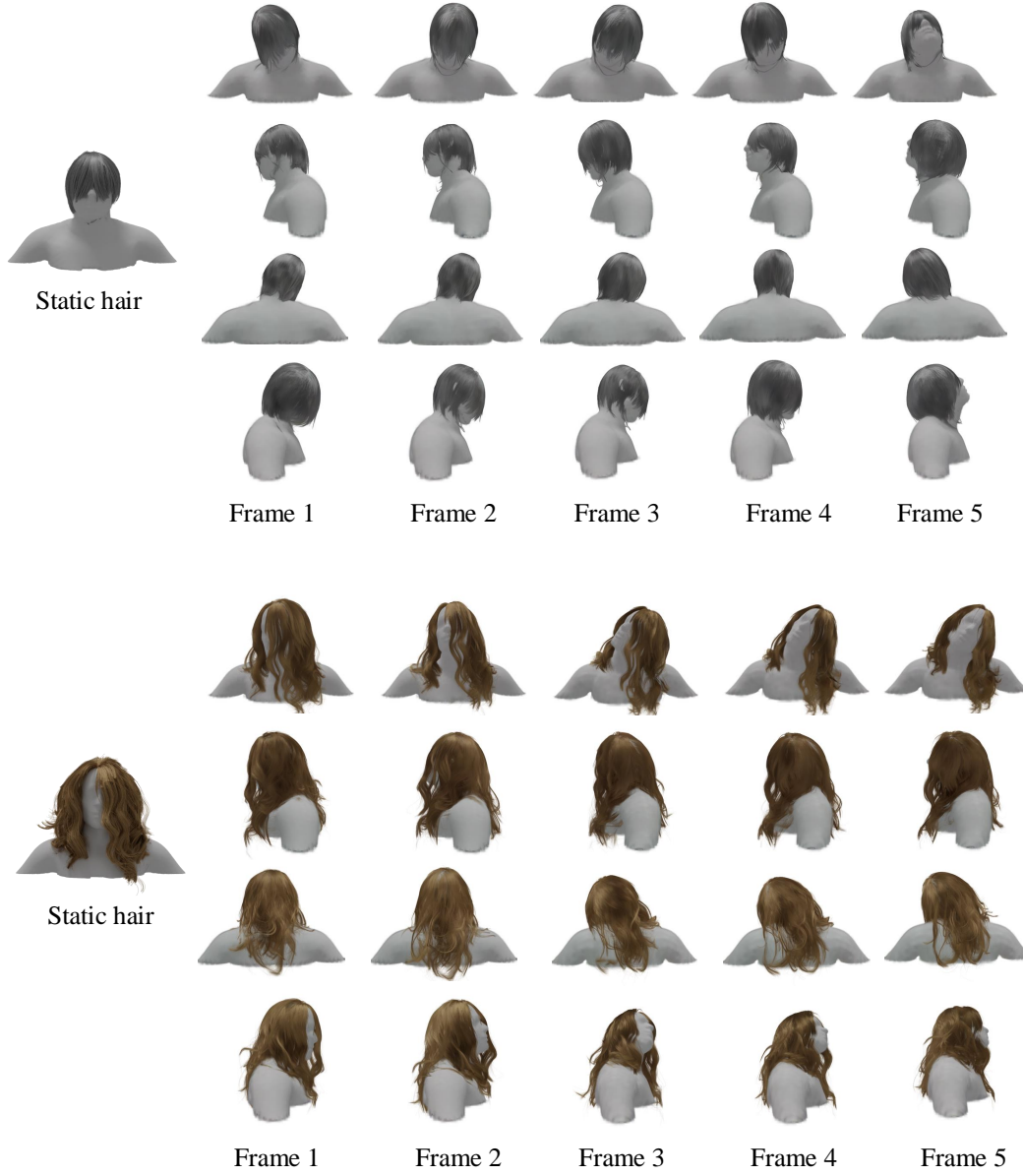


Figure 12. Our dynamic Gaussian hair appearance model enables novel view synthesis from static hair using our hair tracking model. Note: Static hair lacks gravity, while the dynamic model implicitly learns gravity from the dataset distribution.

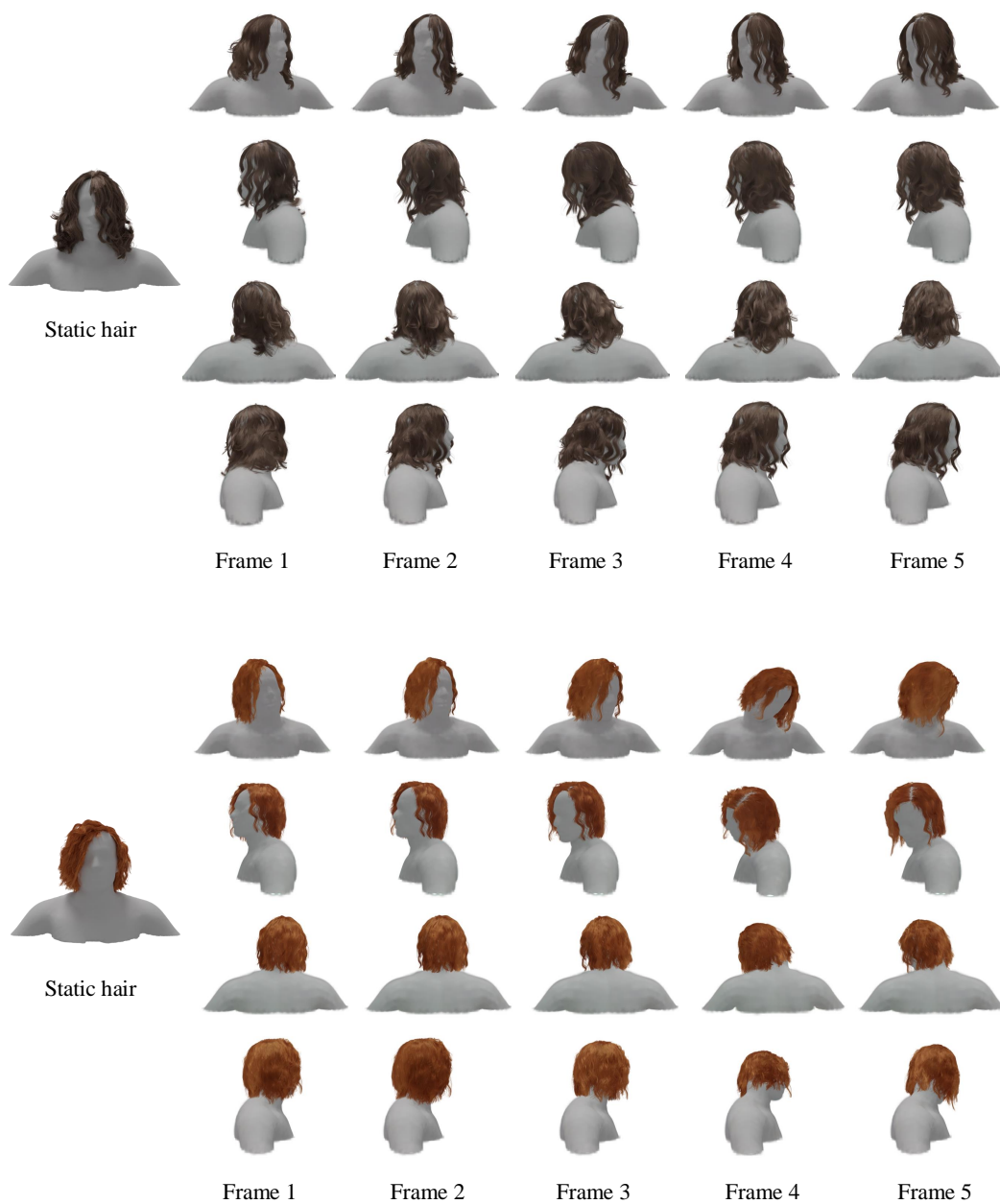


Figure 13. Novel view synthesis results from static hair using our hair tracking model across various hair grooms.

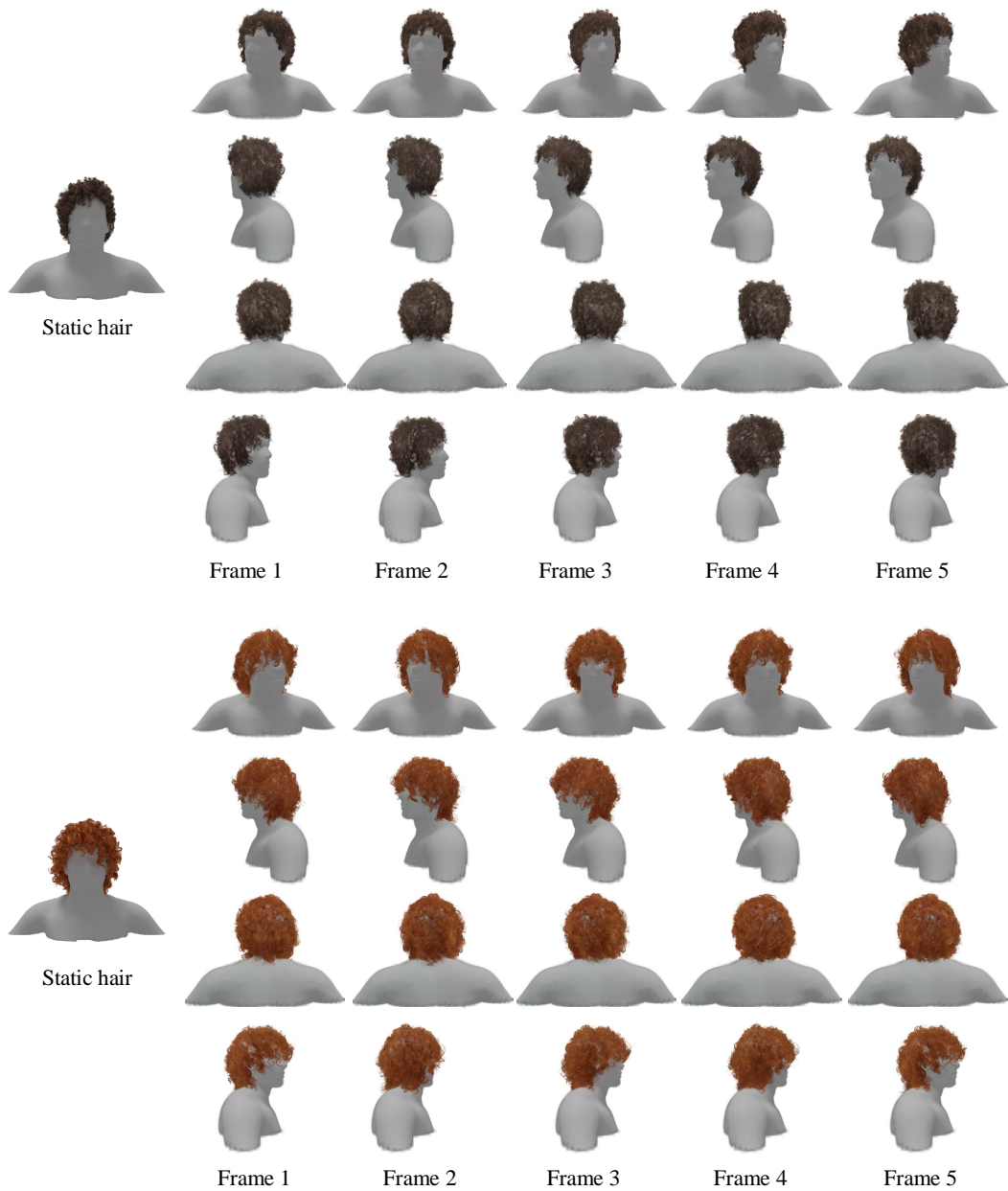


Figure 14. Given our hair tracking model and body pose, our framework enables reanimation and novel-view rendering with hair dynamics, including curly hair.

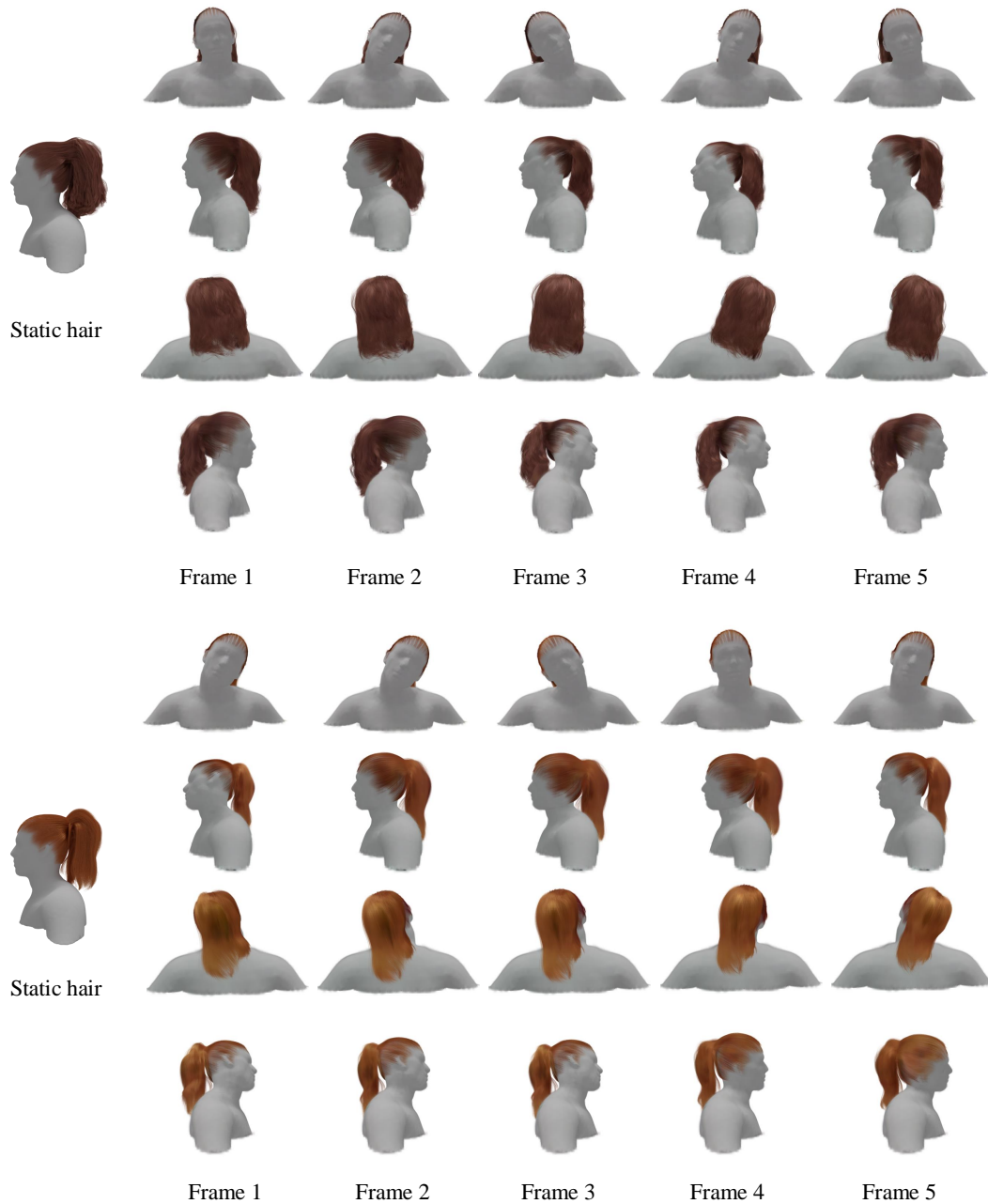


Figure 15. Given our hair tracking model and body pose, our framework enables reanimation and novel-view rendering with hair dynamics, including ponytail hairstyles.

References

- [1] NVIDIA Corporation. *NVIDIA TensorRT: High-Performance Deep Learning Inference*, 2025. URL <https://developer.nvidia.com/tensorrt>. Accessed: 2025-03-05. 5
- [2] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14346–14355, 2021. 8
- [3] Chengan He, Jorge Alejandro Amador Herrera, Yi Zhou, Zhixin Shu, Xin Sun, Yao Feng, Sören Pirk, Dominik L Michels, Meng Zhang, Tuanfeng Y Wang, and Holly Rushmeier. Digital salon: An ai and physics-driven tool for 3d hair grooming and simulation. *ACM SIGGRAPH Asia 2024 Real-Time Live!*, 2024. URL <https://digital-salon.github.io/>. 8
- [4] Chengan He, Xin Sun, Zhixin Shu, Fujun Luan, Sören Pirk, Jorge Alejandro Amador Herrera, Dominik L Michels, Tuanfeng Y Wang, Meng Zhang, Holly Rushmeier, and Yi Zhou. Perm: A parametric representation for multi-style 3d hair modeling. *arXiv preprint arXiv:2407.19451*, 2024. 8
- [5] 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. 7
- [6] Xueting Li, Ye Yuan, Shalini De Mello, Gilles Daviet, Jonathan Leaf, Miles Macklin, Jan Kautz, and Umar Iqbal. Simavatar: Simulation-ready avatars with layered hair and clothing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26320–26330, 2025. 7
- [7] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016. 1, 5
- [8] 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. 4
- [9] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *arXiv preprint arXiv:2106.12372*, 2021. 8
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4
- [11] Egor Zakharov, Vanessa Sklyarova, Michael Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. Human hair reconstruction with strand-aligned 3d gaussians. In *European Conference on Computer Vision*, pages 409–425. Springer, 2024. 4, 7