

StrandHead: Text to Hair-Disentangled 3D Head Avatars Using Human-Centric Priors

Supplementary Material

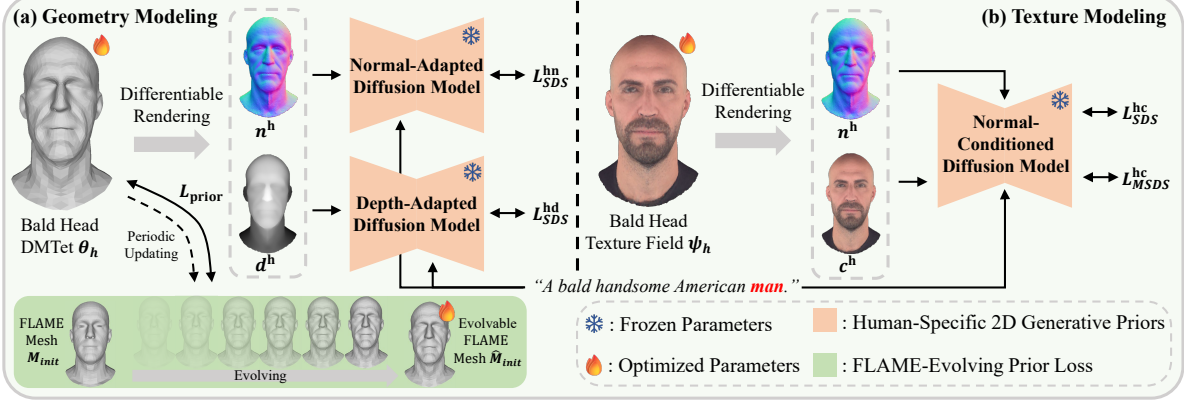


Figure 11. The process for generating a bald head model involves two steps: **(a)** Employing human-specific geometry-aware diffusion models and FLAME-evolving prior loss to model realistic and semantic-aligned bald head shapes. **(b)** Subsequently, using a normal-conditioned diffusion model to generate lifelike head textures.

6. Supplementary Material

This document includes the following supplementary content:

- Bald Head Generation.
- Implementation Details.
- More Evaluations.
- More Experiment Details and Results.
- Prompt List.
- Ethics Statement.

6.1. Bald Head Generation

Following HumanNorm [17] and HeadArtist [30], We use DMTet [60] θ_h and a texture field ψ_h to model the geometry and appearance of the bald head respectively. These components are optimized separately in two stages, as illustrated in Fig. 11.

Head Geometry Modeling. Specifically, we first initialize the head DMTet θ_h utilizing the FLAME model M_{init} . We then refine it under the supervision of human-specific geometry-aware diffusion models (Fig. 11-(a)) by leveraging the following losses:

$$\nabla_{\theta_h} \mathcal{L}_{SDS}^{hn} = \mathbb{E}_{t,\epsilon} \left[(\epsilon_{\phi_{hn}}(n_t^h; y_h, t) - \epsilon) \frac{\partial n^h}{\partial \theta_h} \right], \quad (12)$$

$$\nabla_{\theta_h} \mathcal{L}_{SDS}^{hd} = \mathbb{E}_{t,\epsilon} \left[(\epsilon_{\phi_{hd}}(d_t^h; y_h, t) - \epsilon) \frac{\partial d^h}{\partial \theta_h} \right], \quad (13)$$

where y_h is the input bald head text, ϕ_{hn} and ϕ_{hd} are the human-specialized normal-adapted and depth-adapted diffusion models, respectively, and n^h and d^h are the rendered normal and depth maps of the bald head.

In addition, to obtain semantic information for accurate hair initialization, we introduce a FLAME-evolving prior loss, inspired by Barbie [66]. In specific, we freeze the FLAME shape parameters β but improve M_{init} to evolvable \hat{M}_{init} by adding learnable vertex-wise offsets. We periodically fit \hat{M}_{init} to the head DMTet every δ iteration, thereby obtaining accurate semantic information. Moreover, \hat{M}_{init} provide a reliable and diverse head prior to prevent unnatural geometry using the following formula:

$$\mathcal{L}_{prior} = \sum_{p \in P} \left\| s_{\theta_h}(p) - s_{\hat{M}_{init}}(p) \right\|_2^2, \quad (14)$$

where s_{θ_h} and $s_{\hat{M}_{init}}$ are the signed distance functions (SDF) of the head DMTet θ_h and \hat{M}_{init} , respectively, and P is a set of randomly sampled points. Through this evolving process, \hat{M}_{init} gradually captures rich geometric features (e.g., beards and wrinkles), providing reliable yet diverse priors for subsequent geometry generation (Fig. 12). In summary, the loss function for optimizing the head geometry is as follows:

$$\mathcal{L}_{head-geo} = \mathcal{L}_{SDS}^{hn} + \mathcal{L}_{SDS}^{hd} + \lambda_{prior} \mathcal{L}_{prior}. \quad (15)$$

Head Texture Modeling. Given the generated head geometry generated, we fix it and utilize a texture field ψ_h , which maps a query position to its color to generate head appearance. Specifically, we construct this field using MLP with multi-resolution hash encoding [40], and we optimize it using the following loss function (Fig. 11-(b)):

$$\nabla_{\psi_h} \mathcal{L}_{SDS}^{hc} = \mathbb{E}_{t,\epsilon} \left[(\epsilon_{\phi_{hc}}(c_t^h; n^h, y_h, t) - \epsilon) \frac{\partial c^h}{\partial \psi_h} \right], \quad (16)$$

where ϕ_{hc} is a human-specialized normal-conditioned diffusion model, and c^h represents the rendered color image of the generated head. Since the vanilla SDS loss often leads to color oversaturation, we replace it with the following MSDS loss [17] to further enhance the texture’s realism in later iterations of texture optimization:

$$\begin{aligned} \nabla_{\psi_h} \mathcal{L}_{MSDS}^{hc} = & \mathbb{E}_{t,\epsilon} \left[(h(c_t^h; n^h, y_h, t) - \epsilon) \frac{\partial c^h}{\partial \psi_h} \right] + \\ & \mathbb{E}_{t,\epsilon} \left[(V(h(c_t^h; n^h, y_h, t)) - V(\epsilon)) \frac{\partial V(\epsilon)}{\partial \epsilon} \frac{\partial c^h}{\partial \psi_h} \right], \end{aligned} \quad (17)$$

where V is the first k layers of the VGG network [63] $h(\cdot)$ denotes the multi-step image generation function of the normal-aligned diffusion model.

6.2. Implementation Details

Hyper-Parameters. For the bald head generation, our approach requires 10,000 iterations for geometry creation, 2,000 iterations for texture synthesis, and 5,000 iterations for appearance refinement using MSDS [17]. The loss weight λ_{prior} is set to 1×10^3 . For hair generation, the process involves 5,000 iterations for geometry modeling, 2,000 iterations for texture generation, and 5,000 iterations for visual enhancement using MSDS [17]. The respective loss weights for λ_{SDS}^{hn} , λ_{ori} , λ_{cur} , λ_{bbox} , λ_{face} , and λ_{colli} are assigned as 1×10^{-3} , 1×10^4 , 1×10^4 , 1×10^3 , 1×10^3 , and 1×10^3 respectively. The number of strand polylines N_s and strand points N_p are configured as 3000 and 100, respectively. For different hairstyles—normal, straight, wavy, and curly—the target curvature C_{target} is defined as 5×10^{-2} , 2×10^{-2} , 1×10^{-1} , and 2×10^{-1} , respectively. All experiments are conducted on an Ubuntu server equipped with A6000 GPUs. Generating a bald head takes roughly 4 hours with 24GB of memory, and generating a haircut takes roughly 4 hours with 44GB of memory.

Details of FLAME-Evolving Prior Loss. In the FLAME-evolving prior loss, we periodically optimize \hat{M}_{init} to fit the current human head geometry at every 1000 iterations, thus providing an effective and flexible human head prior constraint. The fitting loss function is as follows:

$$\mathcal{L}_{fit} = \lambda_{chamf} \mathcal{L}_{chamf} + \lambda_{edge} \mathcal{L}_{edge} + \lambda_{nor} \mathcal{L}_{nor} + \lambda_{lap} \mathcal{L}_{lap}, \quad (18)$$

where \mathcal{L}_{chamf} is the Chamfer distance between \hat{M}_{init} and the current human head geometry, \mathcal{L}_{edge} is the edge length regularization loss, \mathcal{L}_{nor} is the normal consistency loss, and \mathcal{L}_{lap} is the Laplacian smoothness loss. The loss weights λ_{chamf} , λ_{edge} , λ_{nor} , and λ_{lap} are set to 1×10^2 , 1×10^0 , 1×10^{-2} , and 1×10^{-1} , respectively. The ablation study results for the FLAME-evolving prior loss are presented in Fig. 12. As shown, excluding \mathcal{L}_{prior} results in exaggerated head shapes (e.g., overly pointed head tops), significantly compromising the realism of the outputs. Employing a non-evolving prior

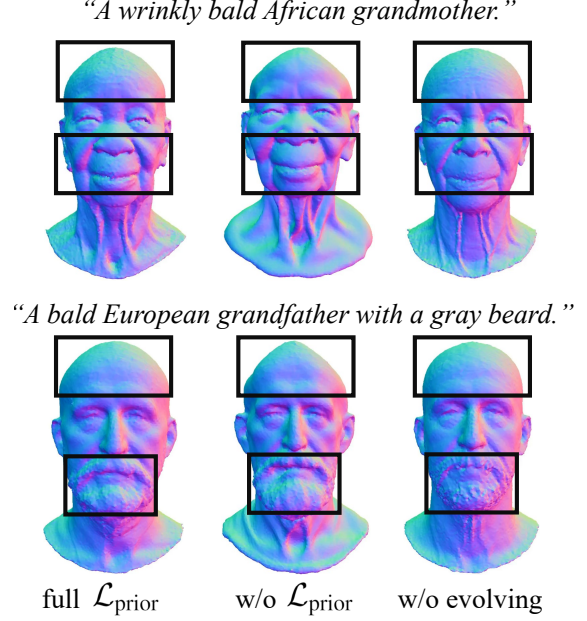


Figure 12. Ablation study on FLAME-evolving prior loss.



Figure 13. Initialize hairstyles. Since the USC-HairSalon Dataset [15] lacks afro hairstyles, we use HAAR [65] to generate the initial afro hairstyle.

loss achieves reasonable head proportions but fails to preserve finer geometric details such as beards and wrinkles. By leveraging the full \mathcal{L}_{prior} , our approach generates head geometry with both accurate proportions and high detail fidelity.

Details of Hair Initialization. As mentioned in the main paper, we utilize ChatGPT[43] to select the most representative hairstyles from the USC-HairSalon Dataset [15]. Specifically, we first exclude some exaggerated hairstyles (e.g., those that are overly long or excessively messy). Next, ChatGPT is used to generate textual descriptions for each

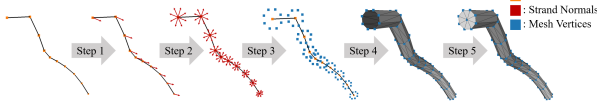


Figure 14. Illustration of converting a hair strand into a prismatic mesh using the differentiable prismatization algorithm.

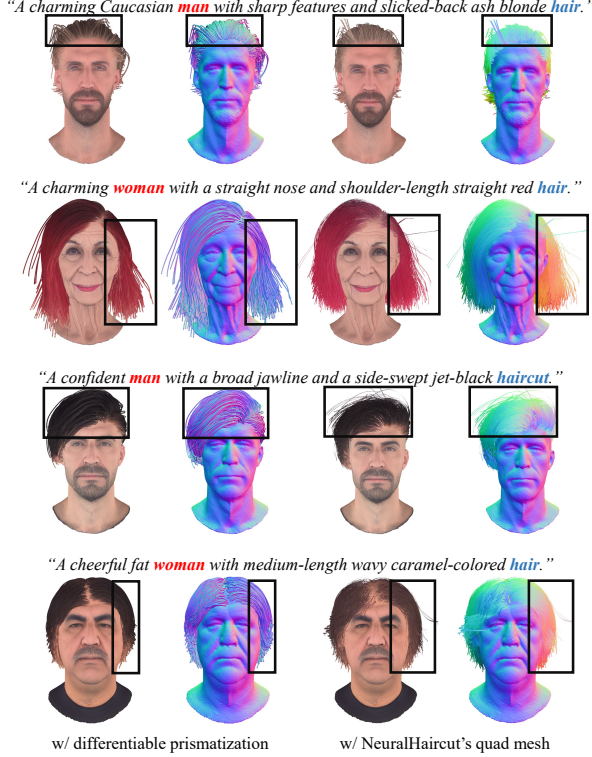


Figure 15. Differentiable prismatization vs. NeuralHaircut’s quad mesh [64].

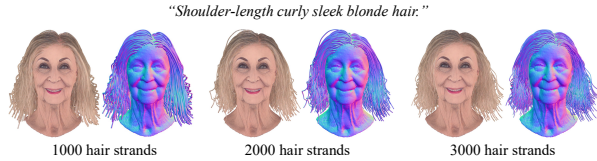


Figure 16. Ablation study on the number of hair strands.

hairstyle, which are then utilized to select the 20 most representative ones. As shown in Fig. 13, the selected hairstyles cover different lengths, curvatures, and styles, ensuring rich diversity. Finally, we optimize neural scalp textures (NST) to fit the selected hairstyle using Eq. (2), where λ_{ori} and λ_{cur} are set 5×10^{-2} , and 1×10^0 , respectively.

Details of Differentiable Prismatization Algorithm. Given a hair strand s , our differentiable prismization algorithm converts it into a watertight prismatic mesh with K lateral edges and radius R through the following five steps:

1. **Compute the Initial Normal Vector:** Determine a normal to the hair strand s by taking the cross product of its orientation with a non-collinear reference point (typically the center of the head).

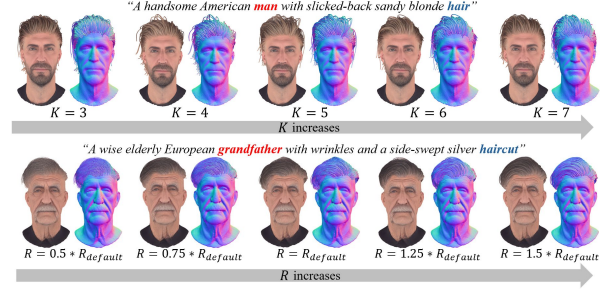


Figure 17. Ablation study on K and R .

2. **Generate K Rotated Normals:** Rotate this normal around the axis defined by the strand’s orientation K times, each by $\frac{360^\circ}{K}$, to produce K normals.
3. **Translate to Form Lateral Edges:** Translate s along each of these K normals by R , generating K lateral edges of the prism.
4. **Construct Lateral Faces:** Connect the adjacent lateral edges’ vertices to form the K lateral faces of the prism.
5. **Construct Top and Bottom Faces:** Connect the vertices at the ends of the lateral edges to form the top and bottom faces of the prism, completing the conversion from a hair strand to a watertight prismatic mesh.

Figure 14 shows an example of converting a hair strand into a prismatic mesh. Importantly, the proposed differentiable prismatization algorithm can be easily implemented on GPU, achieving flexible and fast prismatization of hair strands and paving a new way for hair modeling.

Specific to our experiment, each hair strand is converted into a watertight prismatic mesh with $K = 4$ lateral edges. The radius is defined as $R = \sqrt{\frac{A_{\text{scalp}}}{N_s \pi}}$, where A_{scalp} represents the surface area of the scalp mesh. During the optimization of hair textures, the radius R is further reduced to $\frac{\sqrt{A_{\text{scalp}}}}{2}$ to achieve a more detailed appearance. As illustrated in Fig. 15, our proposed differentiable prismatization algorithm offers more stable gradient backpropagation compared to the quad mesh used by NeuralHaircut [64].

This mitigates abnormal normal problems caused by non-watertight meshes, ensuring reliable strand-based hair optimization. Additionally, we present an ablation study on the effect of the number of hair strands in Fig. 16. It is observed that as the number of hair strands increases, the generated hairstyles become denser. Meanwhile, the overall shape and texture quality are maintained, demonstrating the robustness of differentiable prismatization. As shown in Fig. 17, we also conduct an ablation study on lateral edges K and the radius R . The results indicate that the number of lateral edges K has a minimal impact on the final output, which validates the effectiveness and robustness of our differentiable prismatization algorithm.

Details of Geometry-Aware Losses. The geometry-aware

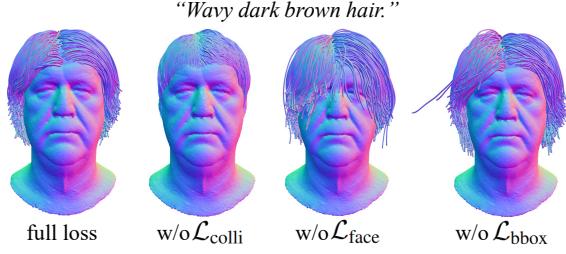


Figure 18. Ablation study on geometry-aware losses.

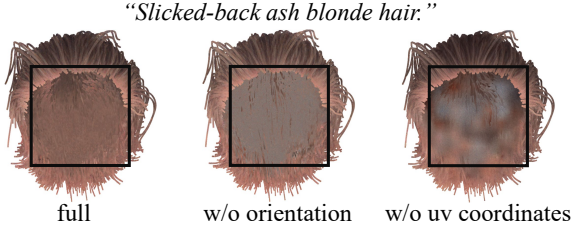


Figure 19. Ablation study on the strand-aware texture field.

loss functions $\mathcal{L}_{\text{bbox}}$, $\mathcal{L}_{\text{face}}$, and $\mathcal{L}_{\text{colli}}$ are formulated as:

$$\mathcal{L}_{\text{bbox}} = \sum_{p \in S} \max(0, s_{\text{bbox}}(p)), \quad (19)$$

$$\mathcal{L}_{\text{face}} = \sum_{p \in S} \max(0, -s_{\text{face}}(p)), \quad (20)$$

$$\mathcal{L}_{\text{colli}} = \sum_{p \in S} \max(0, -s_{\text{head}}(p)), \quad (21)$$

where s_{bbox} , s_{face} , and s_{head} are the SDF of the bounding box, the space in front of the face, and the head, respectively. Here, p represents the 3D points of hair strands S .

The results of the ablation study on geometry-aware losses are displayed in Fig. 18. Incorporating $\mathcal{L}_{\text{bbox}}$, $\mathcal{L}_{\text{face}}$, and $\mathcal{L}_{\text{colli}}$ effectively prevents the hair from extending beyond the bounding box, obscuring the face, and colliding with the head. This significantly enhances the geometric rationality and realism of the generated hairstyles.

Details of Strand-Aware Texture Field. Due to the complexity of hair textures, a basic texture field cannot fully capture the lifelike appearance of strands. For a query point p , the basic texture field ϕ_b generates its color using the following formula:

$$c = \phi_b(\text{Euc}(p)), \quad (22)$$

where $\text{Euc}(\cdot)$ represents the Euclidean coordinates of the query point. To better model high-frequency color variations, we propose the strand-aware texture field ϕ_s which uses the following equation:

$$c = \phi_s(UV(p), o), \quad (23)$$

where $UV(\cdot)$ denotes the scalp UV coordinates of the query point, and o refers to its strand orientation.

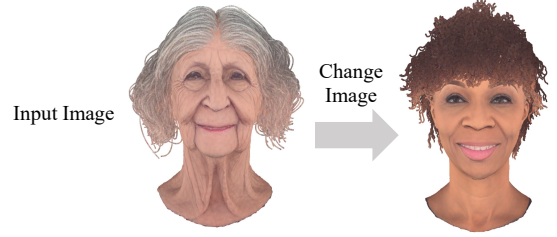


Figure 20. Quantitative comparisons for metrics including CLIP, Open-CLIP, Fashion-CLIP, BLIP-VQA, and BLIP2-VQA.

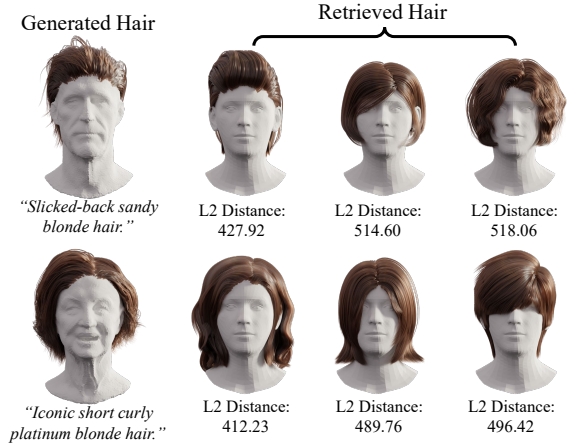


Figure 21. The generated hair and the retrieved hair from USC-HairSalon [15] by ranking the L2 distance of neural scale textures.

Specifically, we introduce two improvements to the basic texture field: first, we replace Euclidean coordinates with scalp UV coordinates, which are more uniformly distributed. Second, we incorporate strand orientations as additional input information to model orientation-dependent texture variations. These enhancements enhance the realism of the generated results by better capturing high-frequency appearance variations and ensuring consistent colors across different faces of a single prismatic mesh since the input features are strand-based. As shown in Fig. 19, our proposed strand-aware texture field accurately models high-frequency appearance details by switching coordinate spaces and incorporating orientation information, resulting in more realistic strand textures.

Method	FID ↓	CLIP ↑
HeadArtist	201.35	27.69
HeadStudio	271.37	27.61
HumanNorm	211.65	27.96
TECA	<u>178.79</u>	<u>30.60</u>
StrandHead (Ours)	176.95	30.95
MVDream	215.06	28.11
GaussianDreamer	249.96	25.05
LucidDreamer	231.65	25.05
RichDreamer	<u>213.54</u>	27.13
TECA	231.23	26.36
HAAR	335.07	25.42
StrandHead (Ours)	201.19	<u>27.84</u>

Table 3. Quantitative comparisons with the SOTA methods. The best and second-best results are highlighted in **bolded** and underlined, respectively.

6.3. More Evaluations

For a more comprehensive comparison, we also provide evaluations based on FID and CLIP metrics. We compute the FID between the images rendered by each method and those generated by Flux. As shown in Tab. 3, our method still achieves good results. However, given the inherent flaws of these metrics, we recommend that readers refer to the metrics in the main paper.

6.4. More Experiment Details and Results

Disadvantages of CLIP-Based Metrics. Recent studies [16, 31] have demonstrated that CLIP-based metrics are limited to assessing coarse text-image similarity and struggle to capture the fine-grained correspondence between 3D content and input prompts accurately. To address this limitation, we follow Progressive3D [7] and utilize fine-grained text-to-image evaluation metrics, such as BLIP-VQA [23, 24] and BLIP2-VQA [23, 25], to assess the generative capabilities of different methods.

As depicted in Fig. 20, when input images are replaced while keeping the input text unchanged, interesting observations can be made: Open-CLIP and Fashion-CLIP scores increase under these conditions. However, BLIP-VQA and BLIP2-VQA scores drop significantly. This highlights the limitations of CLIP-based metrics in evaluating fine-grained correspondences. In contrast, BLIP-VQA and BLIP2-VQA demonstrate superior performance in capturing these intricate relationships.

Clarification on Retrieval-Like Results. Since there is no GT available for text-driven hair generation, we display the generated hair alongside its top-3 nearest haircuts from the USC-HairSalon dataset [15] textures, as shown in Fig. 21. The similarity is ranked based on the L2 distance of neural scale textures. As illustrated, the generated hair is significantly different from the retrieved hair samples. This distinction clearly demonstrates that our method generates unique hairstyles rather than simply retrieving them from the dataset.

More Experiment Results. We present additional 3D hair-disentangled head avatars in Fig. 22 and 3D strand-based

hair generated by StrandHead in Fig. 23.

Effect of Human-Specific 2D Generative Priors. We demonstrate the importance of human-specific 2D generative priors from two aspects:

(1) Fig. 24 displays optimized hair under varying text conditions, while keeping the initial hairstyle and bald head constant. As shown, our method, leveraging human-specific 2D generative priors, accurately captures subtle changes in textual descriptions (e.g., variations in haircut length and curliness). The generated 3D strand-based hair not only exhibits a realistic and well-structured shape but also aligns closely with the given text conditions.

(2) Fig. 25 illustrates generated hair under varying bald head conditions while maintaining a fixed initial hairstyle and hair prompt. As observed, thanks to the powerful 2D diffusion models pre-trained on human data, the generated 3D hair strands exhibit geometry and texture variations that adapt to specific bald heads.

In summary, our robust optimization strategy ensures that the generated hair is not only highly consistent with the text prompts but also integrates harmoniously with the human head.

6.5. Prompt List

The following are textual prompts for quantitative experiments:

- A beautiful girl with delicate features and long, silky black wavy hair.
- A cheerful fat European man with a short, spiky light brown haircut.
- A confident black woman with a curly dark brown afro.
- A handsome American man with slicked-back sandy blonde hair.
- A kind grandmother with big ears and a silver curly bob hairstyle.
- A lively European boy with tousled light brown hair.
- A lively black girl with tight, curly dark brown hair.
- A mature African man with deep-set eyes and a short, curly black afro.
- A mature European woman with a straight nose and medium-length, straight auburn hair.
- A middle-aged Hispanic woman with a strong jawline and medium-length, wavy magenta hair.
- A muscular European man with a wide forehead and slicked-back black hair.
- A serious white man with a sharp nose and a slicked-back, jet-black hairstyle.
- A sexy woman with full lips and long, wavy chestnut brown hair.
- A strong American man with a gray beard and a curly silver haircut.
- A strong man with a broad nose and a short, spiky dark brown haircut.

- A strong man with a strong jaw and a medium-length, wavy black hairstyle.
- A stylish African woman with a sleek, shoulder-length black bob haircut.
- A thin man with a sharp jawline and a spiky light brown mohawk hairstyle.
- A wise elderly European grandfather with wrinkles and a side-swept silver haircut.
- A wise, elderly grandfather with a classic short curly white haircut.
- An elderly African grandmother with wrinkles and a curly gray afro.
- An elderly Caucasian grandmother with thin lips and a straight, short silver bob.
- Angelina Jolie with long, wavy chestnut brown hair.
- Beyoncé with voluminous, curly honey blonde hair.
- Emma Watson with a short, wavy brown bob.
- Lionel Messi with a blonde mohawk hairstyle.
- Michael Jordan with a short, neatly-trimmed black buzz cut.
- Morgan Freeman with a curly brown afro.
- Rihanna with a sleek, long black hairstyle.
- Taylor Swift with long, straight platinum blonde hair.

6.6. Ethics Statement

StrandHead provides an efficient solution for creating realistic 3D head avatars with strand-based hair using 2D/3D human-centric priors, enabling a wide range of applications. However, like many AI generative technologies, it carries the risk of misuse, such as the creation of misleading avatars. To mitigate these concerns, future research in generative AI should emphasize ethical considerations, develop effective safeguards, and promote responsible practices. By addressing these challenges, developers can reduce potential harm while maximizing the positive impact of these technologies.

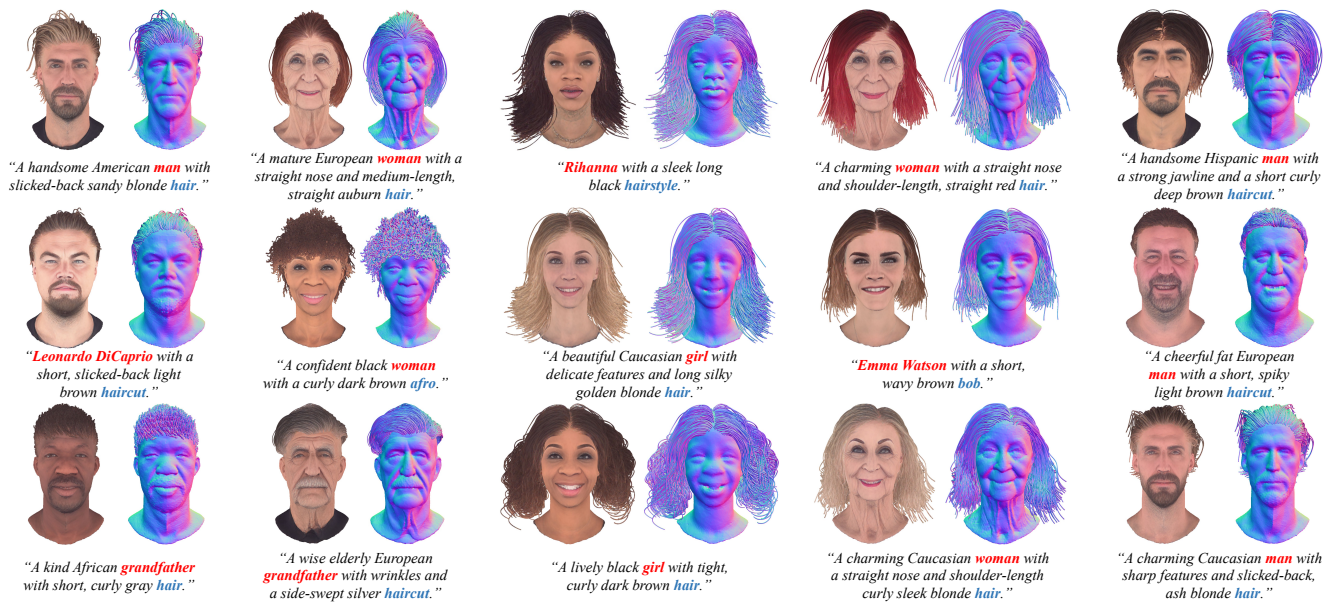


Figure 22. Examples of 3D head avatars with strand-based attributes.



Figure 23. The physics-based hair strand rendering result using Blender [10].

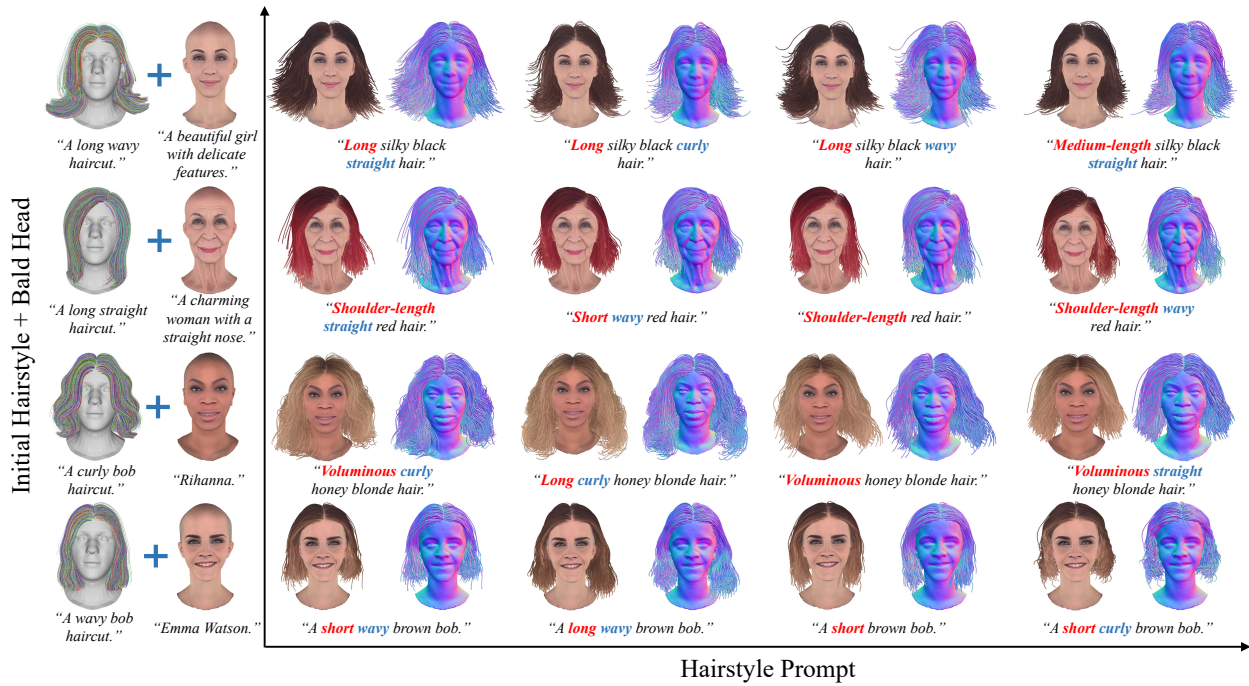


Figure 24. The optimization results under different hair prompts. Starting from the same initial hairstyle, StrandHead demonstrates its capability to generate diverse hairstyles by adapting to varying text conditions.

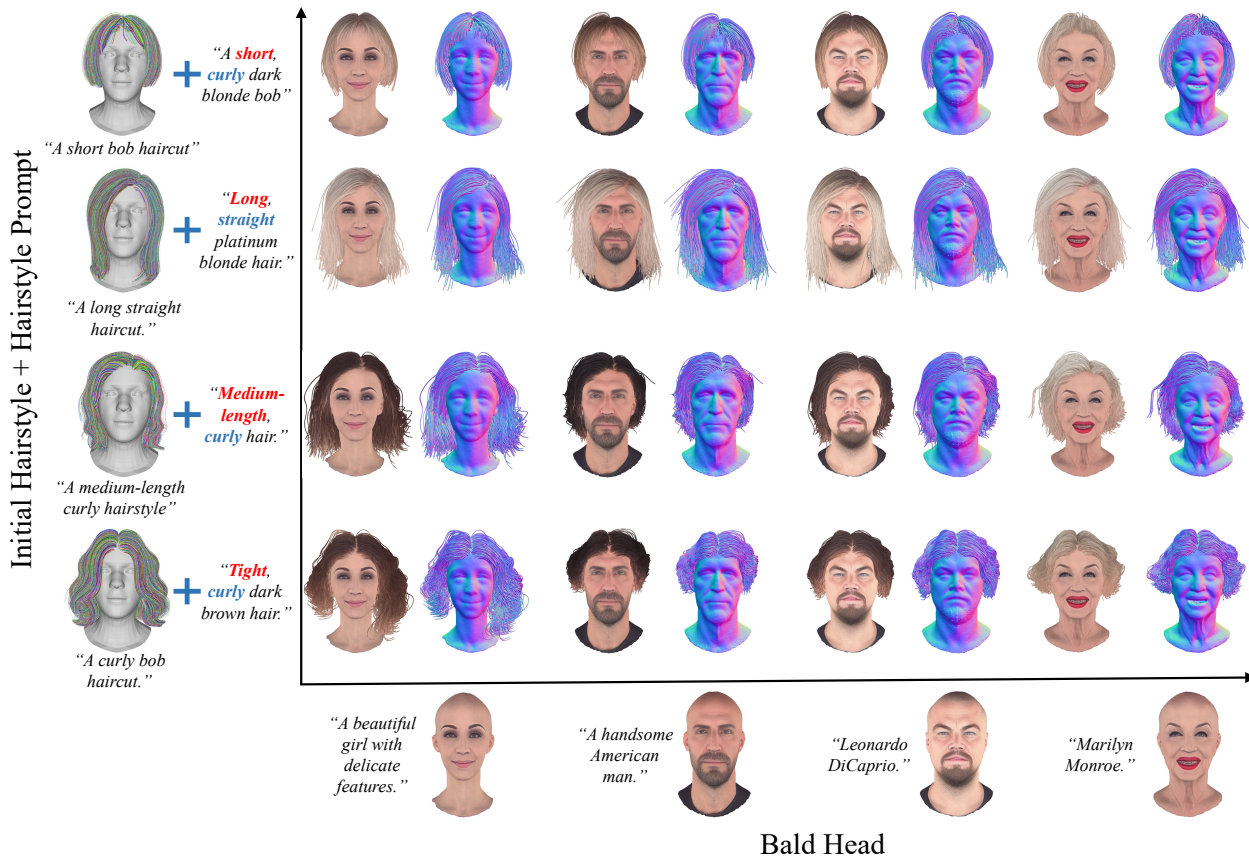


Figure 25. The optimization results under different bald heads. Under consistent text conditions, the 3D hair generated by StrandHead exhibits specific geometry and texture variations that seamlessly adapt to the unique features of different bald heads.