SAFE-Hair: Scalp-Anchored Fields for Exportable Single-View Hair Reconstruction

Yucheng Wang* **Zedong Wang*** **Yuetong Wu** **Yue Ma** **Dan Xu**✉
The Hong Kong University of Science and Technology
{ywangls, zwangmw, ywufe, ymacn}@connect.ust.hk danxu@cse.ust.hk



Figure 1: From a single portrait image, **SAFE-Hair** reconstructs an explicit strand-based 3D hair asset with stable scalp-rooted follicles. We visualize each reconstruction from front, side, and back views, showing complete hairstyle recovery beyond the visible input region. The bottom row highlights the diversity of generated assets across multiple hairstyle categories.

Abstract

Single-view 3D hair reconstruction has progressed rapidly, but many methods optimize for visual plausibility rather than producing strand assets that can be directly used in grooming, editing, or simulation pipelines. We propose **SAFE-Hair**, a framework that reconstructs explicit strand grooms from a single portrait by decoding hair from persistent follicles in a canonical scalp-UV domain. Given a fitted template head, SAFE-Hair projects self-supervised image features onto the scalp, uses a conditional Rectified Flow model to generate latent scalp fields, and decodes occupancy, segment directions, length allocation, and total length at a fixed set of scalp follicles. This representation enforces scalp-rooted strand identities by construction and enables direct export with follicle IDs and guide-family metadata. We further introduce attachment, collision, and bending priors to improve geometric validity without per-subject test-time optimization. To evaluate both reconstruction fidelity and asset usability, we introduce **HairBench-3D**, a benchmark built from public 3D hair assets with standardized alignment, rendering, resampling, and metrics for geometry, direction consistency, invisible-region completion, silhouette agreement, and collision. SAFE-Hair improves Chamfer distance from 3.74 mm to

*Equal contribution. ✉Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

3.28 mm, F-score from 77.9 to 81.7, invisible-region F-score from 66.1 to 69.8, and penetration ratio from 4.8% to 3.9% over the strongest baseline. Code is available at this link.

1 Introduction

Reconstructing 3D hair from a single portrait image is an important step toward scalable digital human creation for visual effects, augmented reality, and telepresence [30]. Recent learning-based methods have made substantial progress in recovering visually plausible hair geometry from monocular inputs. However, visual plausibility alone is often insufficient for downstream content-creation pipelines. In practical grooming and simulation workflows, hair is expected to be represented as strands with well-defined roots on the scalp, stable root identities, and limited interpenetration with the head. In this paper, we distinguish between reconstructing hair geometry and reconstructing a hair asset. A hair asset is not only a set of visually plausible curves or surfaces; it also requires roots that remain attached to a valid scalp domain, persistent strand or follicle identities for editing, and a structure that can be serialized into standard groom pipelines. This distinction is important because many downstream operations—such as guide selection, child-strand interpolation, simulation constraints, collision handling, and artist editing—are defined relative to scalp-rooted strand identities. A reconstruction that lacks these identities may look plausible when rendered but can still be difficult to edit, simulate, or transfer across frames and subjects.

A second limitation is evaluation. Existing works are commonly evaluated on different datasets, preprocessing pipelines, and metrics. Distance-based or rendering-based metrics are useful, but they do not fully capture whether a reconstructed hairstyle is structurally valid as a strand asset. For example, a method may achieve a low Chamfer distance while placing roots away from the scalp, producing unstable strand identities, or generating strands that penetrate the head. This makes it difficult to compare methods fairly and to measure progress toward exportable hair reconstruction.

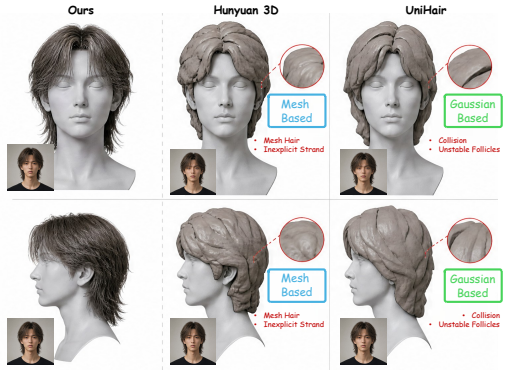


Figure 2: **Motivation.** Existing 3D hair generators often produce rendering-oriented representations: Mesh-based hair [39] lacks explicit strands, while Gaussian-based hair (UniHair [41]) lacks persistent follicles and suffers from collisions. Our method reconstructs explicit rooted strands with stable scalp follicles, enabling editable and export-ready hair assets from a single portrait.

To address these limitations, we propose **SAFE-Hair**, a single-view 3D hair reconstruction framework that produces scalp-indexed strand grooms. Instead of predicting unordered 3D strands directly, SAFE-Hair represents hair in a canonical scalp-UV domain with a fixed set of persistent follicles. Given a portrait image, we fit a template head, warp self-supervised image features into the canonical scalp space, and use a conditional Rectified Flow model [15] to generate latent scalp fields. A shared follicle decoder queries these fields at canonical follicles to predict follicle occupancy, strand directions, and explicit length allocation. This scalp-follicle formulation makes root placement valid with respect to the fitted scalp by construction and gives each decoded strand a persistent identity. We further introduce attachment, collision, and bending priors to reduce off-scalp drift, head interpenetration, and implausible local strand deformation. At inference time, SAFE-Hair reconstructs dense strand assets in a single forward pass without test-time optimization. The persistent follicle representation also provides a natural interface for downstream asset construction. Each decoded strand carries a stable follicle index and can be grouped into guide families after reconstruction, allowing the output to be serialized as a structured strand groom. We treat such guide assignments and groom-control attributes as export metadata rather than the primary reconstruction target, and focus our evaluation on geometric fidelity, root consistency, completion, and collision behavior.

We also introduce **HairBench-3D**, a unified benchmark for evaluating both reconstruction fidelity and asset usability. HairBench-3D is built from public 3D hair resources, including USC-HairSalon [10], Hair20K [43], and public hair models by Yuksel et al. [34]. We standardize these assets into a shared head and scalp coordinate system, render consistent single-view inputs, resample strand curves, and

evaluate methods using metrics for point-level geometry, local direction consistency, invisible-region completion, silhouette agreement, and head-collision behavior.

Our contributions are summarized as follows:

1. We introduce a scalp-anchored representation for single-view 3D hair reconstruction, where strands are decoded from persistent canonical follicles in a shared scalp-UV domain.
2. We propose SAFE-Hair, a conditional Rectified Flow framework over latent scalp fields coupled with a follicle decoder for predicting strand occupancy, directions, and explicit length allocation without test-time optimization.
3. We construct exportable strand grooms with stable follicle identities, scalp-attached roots, collision-aware geometry, and guide-family metadata for downstream editing and simulation.
4. We present HairBench-3D, a unified benchmark with standardized preprocessing and metrics for geometry, direction consistency, invisible-region completion, silhouette agreement, and collision behavior.

2 Related Work

3D hair assets and strand representations. Explicit strand curves are a natural representation for production hair assets, since they directly encode fiber geometry and support rendering, editing, and simulation [1, 4, 5, 22]. Public strand-based datasets, including Yuksel's hair models, USC-HairSalon, and Hair20K, have enabled data-driven reconstruction and generation by providing diverse 3D hairstyles aligned to canonical heads [10, 34, 43]. However, learning directly over unordered dense strands remains challenging due to high dimensionality, variable topology, and complex local structures. Recent representations therefore introduce more structured parameterizations, such as guide-and-residual decompositions or differentiable rendering primitives [7, 12, 17, 36, 44], in line with structured part-level latents for general 3D asset generation [6]. Our work follows the strand-based direction but represents hairstyles through persistent scalp follicles, producing explicit, editable strands that are directly usable as hair assets.

Single-view hair reconstruction. Single-view 3D hair reconstruction is severely ill-posed because much of the hairstyle is self-occluded or ambiguous from one portrait. Early methods address this ambiguity through example-based retrieval, deformation, and user-provided cues [2, 3, 9]. Learning-based methods reduce manual interaction by predicting strand features, orientation fields, or intermediate strand/depth maps from image observations [7, 11, 16, 32, 37, 40, 42]. These methods demonstrate that learned priors can recover plausible monocular hair geometry, but their outputs are often sensitive to image-space cues and may lack stable strand identities, persistent scalp roots, or asset-level validity under occlusion. In contrast, we formulate single-view reconstruction in a canonical scalp-UV domain with fixed follicles, explicitly tying strands to persistent scalp anchors.

Multi-view reconstruction and generative priors. Multi-view images or monocular videos provide stronger geometric constraints and have enabled high-fidelity strand reconstruction with learned priors, implicit fields, differentiable rendering, or image-based optimization [13, 18, 23, 25, 27–29, 33, 36, 38, 45]. Recent generative models further use diffusion or flow-based formulations to learn expressive hairstyle priors for image-conditioned 3D hair generation [8, 14–16, 26, 41, 44]. Beyond hair, diffusion models also underpin controllable visual generation and editing, including pose-guided and relightable video generation [19–21] and contextual image editing [31]. While effective, these approaches often rely on multi-view inputs, per-subject optimization, or rendering. Our goal is complementary: reconstruct production-oriented explicit strand assets from a single portrait, with stable follicle indexing and evaluation criteria that measure asset validity.

3 Method

Given a single RGB portrait I, SAFE-Hair reconstructs an explicit strand-based hair asset

$$\mathcal{H} = (\mathcal{S}, \mathcal{R}, \mathcal{E}), \tag{1}$$

where $\mathcal{S}$ denotes the reconstructed 3D strands, $\mathcal{R}$ denotes persistent follicle identities on a canonical scalp, and $\mathcal{E}$ denotes optional export metadata such as guide-family assignments and groom-control

descriptors. Our objective is to recover a visually plausible hairstyle while producing an asset with valid scalp-rooted strands, stable follicle correspondence, and reduced head interpenetration. SAFE-Hair follows a scalp-indexed design. Instead of predicting strands directly in image or world coordinates, we first lift image evidence to a canonical scalp domain, generate dense scalp-aligned latent fields with Rectified Flow [14, 15], and then decode explicit strands from a fixed set of persistent follicles. This design makes root placement valid by construction, keeps strand identities stable across examples, and produces curve-based assets that can be exported without test-time optimization.

3.1 Overview

Shown in Figure 3, SAFE-Hair operates in a canonical scalp parameterization shared by all subjects. Given an input portrait, we first fit a template head and use it as an approximate geometric bridge between the image, the 3D head, and the canonical scalp domain, following common practice in monocular head and hair reconstruction [27]. Sec. 3.2 introduces this representation, including the persistent follicle layout and the projection of dense image features into scalp UV space. Conditioned on the resulting scalp-aligned feature map, SAFE-Hair generates latent hairstyle fields with the conditional Rectified Flow model described in Sec. 3.3. These fields are decoded into explicit rooted strands by the shared follicle decoder in Sec. 3.4. Training objectives, including flow matching, strand-level supervision, image-space constraints, and geometric-validity priors, are summarized in Sec. 3.5. Finally, Sec. 3.6 describes optional groom-control descriptors and guide-family assignments for export. Because every active strand remains tied to a fixed follicle identity, the reconstructed asset naturally supports stable correspondence and standard curve-based groom serialization.

3.2 Canonical scalp space and image conditioning

We assume a template head mesh $\mathcal{M}$ with a scalp region $\mathcal{M}_{\text{scalp}}$ and a canonical UV parameterization $\psi : \mathcal{U} \to \mathcal{M}_{\text{scalp}}$. After fitting the template to the input portrait with an off-the-shelf monocular head reconstruction method, all hair reasoning is performed in this canonical scalp space rather than directly in the image plane. A key component of our representation is a fixed set of canonical follicles

$$\mathcal{R} = \{\mathbf{u}_r\}_{r=1}^{R}, \qquad \mathbf{x}_{r,1} = \psi(\mathbf{u}_r), \tag{2}$$

where $\mathbf{u}_r$ is a scalp UV location and $\mathbf{x}_{r,1}$ is its corresponding 3D root position. The follicle set is shared by all examples, so the index r serves as a persistent follicle identity. This choice anchors every decoded strand to a valid scalp location and avoids unconstrained root prediction in 3D space.

To condition the model, we extract dense image features with a self-supervised image encoder and warp them to scalp UV space using the fitted geometry and camera projection. Self-supervised vision features provide robust dense cues for correspondence and reconstruction tasks [24]. The resulting scalp-aligned conditioning map C places visible image evidence in a canonical support, where it can be fused with learned hairstyle priors to complete ambiguous or invisible regions.

3.3 Rectified-flow generation of scalp fields

For each training hairstyle, we first convert its strand geometry into a canonical scalp-aligned supervision field. Each ground-truth strand is assigned to its nearest canonical follicle according to geodesic distance on the scalp. For an occupied follicle, the associated strand is resampled to a fixed number of control points and encoded with its occupancy, local segment directions, total length, and normalized segment-length allocation. These per-follicle quantities are rasterized to a UV grid and compressed into a latent scalp field $z_1 \in \mathbb{R}^{H \times W \times C_z}$. Empty follicles contribute occupancy labels but no geometric targets. Further details of the canonical field construction are provided in Appendix A.2.

We model the distribution of latent scalp fields with Rectified Flow [14, 15]. Given the scalp-aligned conditioning map C, a Gaussian sample $z_0 \sim \mathcal{N}(0, I)$, a training target z_1, and $t \sim \mathcal{U}(0, 1)$, Rectified Flow constructs the interpolation

$$z_t = (1 - t)z_0 + tz_1. \tag{3}$$

We train a conditional vector field v_θ to predict the transport direction from z_0 to z_1:

$$\mathcal{L}_{\text{rf}} = \mathbb{E}_{z_0, z_1, t}\left[\|v_\theta(z_t, t, C) - (z_1 - z_0)\|_2^2\right]. \tag{4}$$

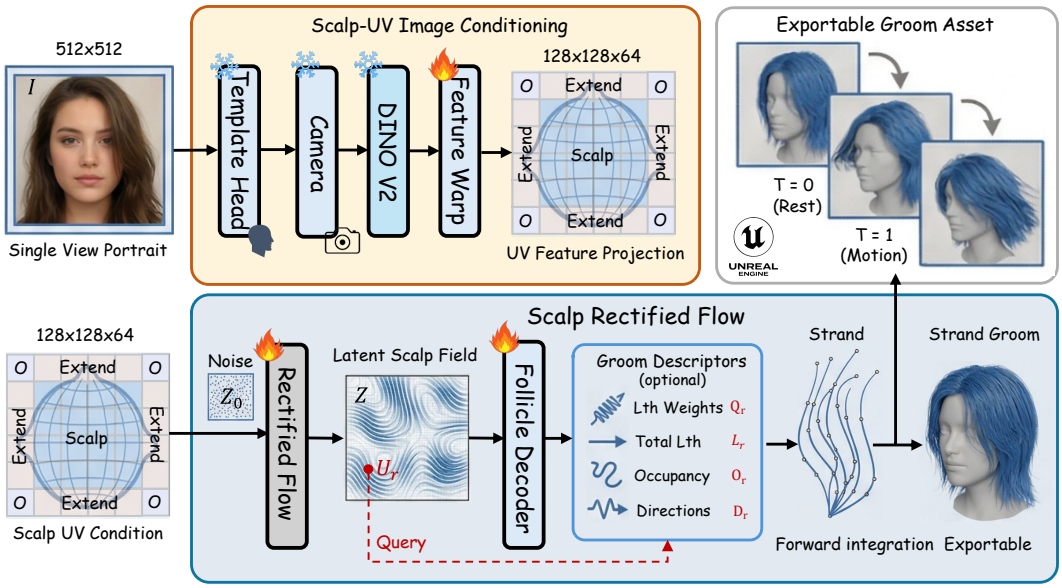


Figure 3: **Overview of SAFE-Hair.** Given a single portrait, we fit a template head and use the recovered camera geometry to project dense image features into a canonical scalp-UV domain, producing a scalp-aligned conditioning map. A conditional Rectified Flow model transforms Gaussian noise into a latent scalp field Z. A shared follicle decoder queries Z at persistent canonical follicles $\{\mathbf{u}_r\}_{r=1}^{R}$ and predicts occupancy, segment directions, length allocation, total length, and export descriptors. The final strand curves are obtained by forward integration from fixed scalp roots, yielding explicit scalp-rooted strands with stable follicle identities and groom metadata for downstream export.

At inference time, we sample $z_0 \sim \mathcal{N}(0, I)$ and integrate

$$\frac{dz_t}{dt} = v_\theta(z_t, t, C), \qquad Z = z_{t=1}, \tag{5}$$

using a fixed-step ODE solver. Unless otherwise specified, a single sample is used for all evaluations.

3.4 Persistent follicle decoding

Given the generated latent scalp field Z, we decode one candidate strand at each canonical follicle. For follicle r, we extract a local feature

$$\mathbf{h}_r = Q(Z, \mathbf{u}_r), \tag{6}$$

where Q denotes bilinear sampling with local aggregation in scalp UV space. A shared decoder predicts follicle occupancy, strand geometry, and optional groom-control descriptors:

$$(o_r, \mathbf{q}_r, \mathbf{D}_r, L_r, \mathbf{e}_r) = D_\phi(\mathbf{h}_r), \tag{7}$$

where $o_r \in [0, 1]$ is the occupancy probability, $\mathbf{q}_r$ are normalized segment-length weights, $\mathbf{D}_r = \{\hat{\mathbf{d}}_{r,t}\}_{t=1}^{T-1}$ are unit segment directions, L_r is the total strand length, and $\mathbf{e}_r$ is an optional descriptor used only for export metadata. We enforce $\mathbf{q}_r$ with a softmax, normalize predicted directions to unit length, and constrain L_r to be positive with a softplus output.

The strand control points are obtained by forward integration from the scalp root:

$$\ell_{r,t} = L_r q_{r,t}, \qquad \mathbf{x}_{r,t+1} = \mathbf{x}_{r,t} + \ell_{r,t}\hat{\mathbf{d}}_{r,t}. \tag{8}$$

A strand is kept in the final set if o_r exceeds a threshold. This factorization separates root position, total length, and local segment allocation, which makes the reconstructed strand explicitly scalp-rooted and prevents length from being represented only by unconstrained coordinate offsets.

3.5 Training objectives and geometric priors

SAFE-Hair is trained with a combination of generative, strand-level, image-space, and geometric-validity objectives. For 3D hair assets, we supervise follicle occupancy, strand directions, length

allocation, total length, and reconstructed control points in canonical follicle space. For real portraits, we render the reconstructed strands and compare them to image using silhouette and orientation cues:

$$\mathcal{L}_{\text{img}} = \lambda_{\text{sil}}\mathcal{L}_{\text{sil}} + \lambda_{\text{ori}}\mathcal{L}_{\text{ori}}. \tag{9}$$

To improve asset validity, we regularize decoded strands with lightweight attachment, collision, and bending priors. The attachment prior encourages the first segment of a strand to emerge outward from the scalp surface. The collision prior penalizes strand samples that intersect or approach the fitted head geometry too closely. The bending prior discourages abrupt local kinks while still allowing naturally curved hairstyles. These priors reduce root flipping, head interpenetration, and unstable strand shapes without requiring test-time optimization. Optional groom-control descriptors are supervised when corresponding geometry-derived targets or metadata are available. In our implementation, such descriptors may include rest length, curl, or other low-dimensional quantities derived from the reconstructed curve. We treat them as export metadata rather than as the primary target.

The full training objective is

$$\mathcal{L} = \lambda_{\text{rf}}\mathcal{L}_{\text{rf}} + \lambda_{\text{strand}}\mathcal{L}_{\text{strand}} + \lambda_{\text{img}}\mathcal{L}_{\text{img}} + \lambda_{\text{valid}}\mathcal{L}_{\text{valid}} + \lambda_{\text{desc}}\mathcal{L}_{\text{desc}}. \tag{10}$$

Here $\mathcal{L}_{\text{strand}}$ includes occupancy, length, direction, and point-level supervision, while $\mathcal{L}_{\text{valid}}$ groups the attachment, collision, and bending priors. In practice, synthetic or 3D-asset samples provide direct strand supervision, while real portraits mainly contribute image-space supervision together with the shared geometric-validity priors.

3.6 Exportable groom construction

Because each strand is decoded from a canonical follicle $\mathbf{u}_r$, its root identity is the follicle index r, which remains stable across examples and downstream processing. For each active strand, we export its control points, follicle ID, scalp coordinate, and optional groom descriptor $\mathbf{e}_r$, providing a stable interface for editing, correspondence, and serialization.

We optionally construct guide families by selecting guide strands from active follicles and assigning each remaining strand to the nearest guide in a joint scalp-position and strand-feature space:

$$g(r) = \arg\min_{m\in\{1,\dots,G\}} \left[\alpha \left\|\mathbf{u}_r - \mathbf{u}_m\right\|_2^2 + \beta \left\|\mathbf{f}_r - \mathbf{f}_m\right\|_2^2\right], \tag{11}$$

where $\mathbf{f}_r$ is a low-dimensional strand descriptor. The assignment is used only as export metadata and does not affect the reconstructed geometry. The final curves can be serialized into standard groom formats such as Alembic or engine-native strand assets. At inference time, SAFE-Hair produces dense strand assets in a single forward pass without test-time optimization.

4 Experiments

4.1 HairBench-3D and Evaluation Protocol

We evaluate on HairBench-3D, a benchmark for single-view 3D hair reconstruction. HairBench-3D compares methods with different output representations, including strands, meshes, implicit fields, and other 3D hair proxies, under a unified evaluation protocol with identical inputs, camera parameters, head alignment, and metric implementations. Benchmark construction, asset filtering, registration, visibility labeling, metric definitions, and output conversion are detailed in Appendix B.

Held-out test split. HairBench-3D contains 1,551 frozen test hairstyles collected from multiple strand-level sources, including Hair20K, USC-HairSalon, and Yuksel Hair Meshes [10, 34, 43]. The test split is fixed before model development and used only for final evaluation. No hairstyle asset, rendered view, augmented variant, or near-duplicate from the test split is used for training, validation, hyperparameter tuning, data filtering, or qualitative example selection. We describe the asset-level separation and near-duplicate removal procedure in Appendix B.2. We will release the whole **HairBench-3D** together with the metrics script to HuggingFace soon.

Table 1: **HairBench-3D held-out test split.** The split is defined at the hairstyle-asset level.

Source	#Assets	Main role
Hair20K [43]	1,200	Diverse hairstyles
USC-HairSalon [10]	343	Groom assets
Yuksel [34]	8	Varied examples
Total	1,551	Frozen test set

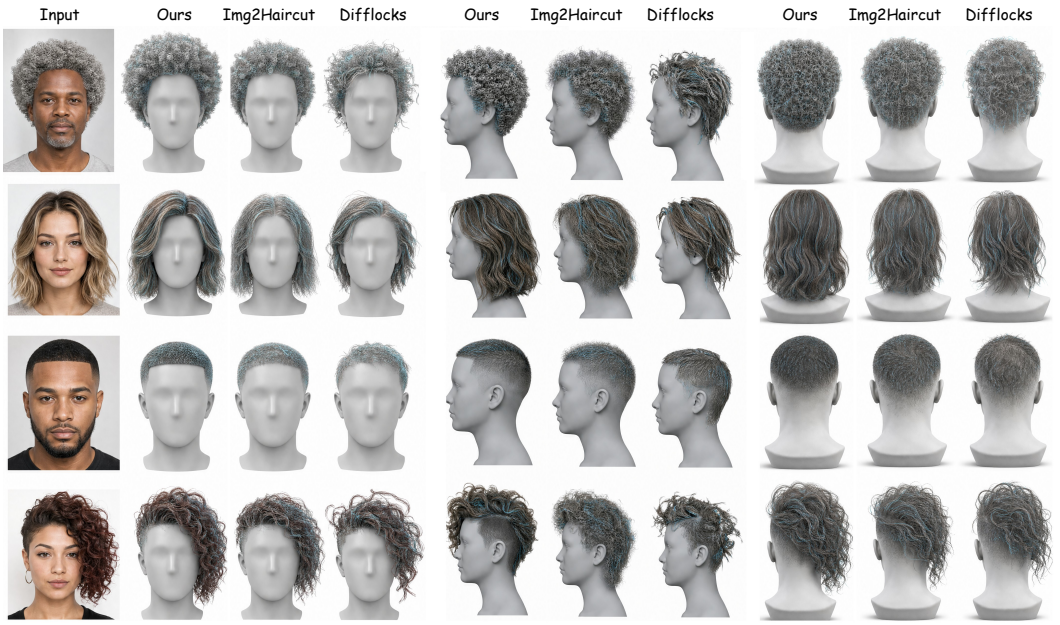


Figure 4: **Qualitative comparison.** For each input image, we show reconstructed hair from multiple viewpoints. SAFE-Hair produces scalp-attached strand geometry with more consistent back-view completion and fewer head-interpenetration artifacts than prior methods.

Method	ChamferPts ↓	F-score ↑	F-score (Invisible) ↑	IoU ↑	Penetration Ratio ↓
Perm [7]	5.82	61.4	42.7	0.781	8.6
UniHair [41]	4.85	69.2	57.0	0.807	26.8
NeuralHDHair [32]	4.36	71.8	56.2	0.824	6.1
Hairstep [40]	4.48	70.6	59.4	0.872	7.2
DiffLocks [26]	3.91	76.5	63.8	0.847	5.4
Im2Haircut [28]	3.74	77.9	66.1	0.861	4.8
Ours	**3.28**	**81.7**	**69.8**	**0.884**	**3.9**
Ours (5 seeds)	3.29 ± 0.03	81.6 ± 0.2	69.7 ± 0.3	0.883 ± 0.001	4.0 ± 0.15

Table 2: **Main results on HairBench-3D.** Lower is better for ChamferPts and Penetration Ratio, while higher is better for F-score@4mm/40°, Invisible-region F-score, and Silhouette IoU. "Ours" uses one fixed seed; "Ours (5 seeds)" reports mean $\pm$ standard deviation over five Rectified Flow sampling seeds with fixed weights.

Canonical evaluation. All assets are registered to a shared template head and scalp-UV parameterization. Each valid strand root is projected to the scalp surface and assigned to a canonical follicle. All methods are evaluated in the same canonical 3D coordinate system, with distance-based metrics reported in millimeters. For each test asset, we render the same calibrated near-frontal RGB input, hair mask, and orientation map for all methods. Held-out side and back views are used only for render-space metrics and are never provided during inference.

Metrics and fairness. We report five primary metrics: ChamferPts, F-score@4mm/40°, Invisible-region F-score, Silhouette IoU, and Penetration Ratio. These metrics measure point-level geometry, tangent-aware reconstruction quality, invisible-region completion, rendered silhouette agreement, and physical validity, respectively. For baselines, we use official pretrained weights and inference code when available. Since prior methods rely on different private, synthetic, or partially released training collections, we enforce fairness through a fixed held-out test split, identical inputs, identical canonical alignment, and representation-independent metric conversion. Formal metric definitions and baseline conversion details are provided in Appendix B.4 and Appendix B.5.

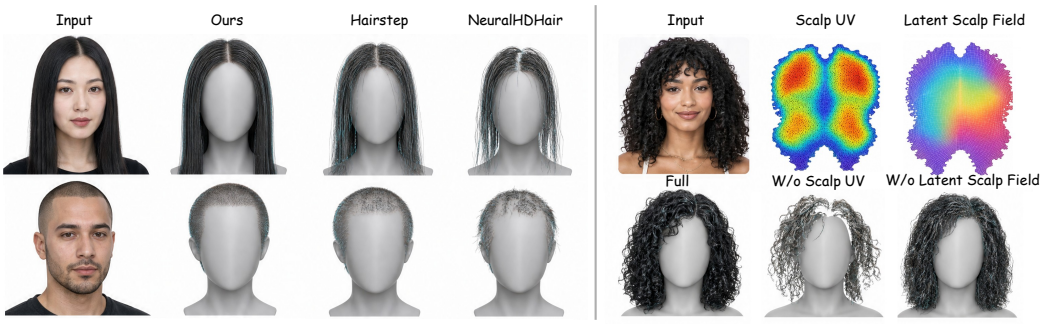


Figure 5: **Qualitative ablation. Left:** SAFE-Hair produces more complete and coherent scalp-rooted strands. **Right:** removing scalp-UV conditioning or the latent scalp field degrades strand coherence and hairstyle consistency.

4.2 Implementation Details

Training data. SAFE-Hair is trained on an independent dataset that contains large-scale synthetic strand renderings and curated high-quality hairstyles. No assets from HairBench-3D are used for training, validation, or hyperparameter selection. Following recent strand-based reconstruction methods [7, 28], our model learns hair priors from synthetic and curated hairstyles. After filtering invalid roots, extremely short strands, and assets with severe scalp misalignment, we split the independent dataset into 18,000 training hairstyles and 1,000 validation hairstyles. HairBench-3D is used exclusively for evaluation, and all benchmark results are reported on its 1,551-style test split.

Model and optimization. We use $N = 32{,}768$ canonical follicles and resample each active strand to $T = 32$ control points. Input images are cropped to 512×512, the scalp feature map has resolution 128×128, and the latent scalp field has $C = 64$ channels. We train for 300k iterations using AdamW with learning rate 2×10^{-4}, weight decay 10^{-4}, batch size 8, and a cosine learning-rate schedule with 10k warm-up iterations. Training is performed on 4 NVIDIA L40 GPUs and takes around 45 hours. Unless otherwise stated, we use $\lambda_{\text{rf}} = 0.1$, $\lambda_{\text{strand}} = 1.0$, $\lambda_{\text{img}} = 0.1$, $\lambda_{\text{valid}} = 0.05$, and $\lambda_{\text{desc}} = 0.01$. Weights are selected on the validation split and fixed for all experiments.

Inference. At inference time, SAFE-Hair reconstructs dense scalp-anchored strands in a single forward pass without test-time optimization. All inference results are generated on a single NVIDIA RTX 3090 GPU. For a 512×512 input image with $32{,}768$ follicles, $T = 32$ control points per strand, and 20 Euler steps, inference takes 4.2 seconds on average, excluding optional file export. The effect of the number of sampling steps and of the output resolution is analyzed in Appendix A.5.

4.3 Baselines

We compare against representative single-view hair reconstruction methods spanning several output representations and inference paradigms, including regression-based strand prediction, implicit reconstruction, hair-growing methods, diffusion-based generation, and optimization-based fitting. Our main comparisons include Perm [7], UniHair [41], NeuralHDHair [32], Hairstep [40], DiffLocks [26], and Im2Haircut [28]. For methods whose native outputs are not dense strands, we convert their results to the common benchmark representation required by each metric. For mesh-based outputs, we uniformly sample surface points and estimate local directions when applicable. For strand-based outputs, we resample strands to the benchmark control-point convention before computing metrics.

4.4 Main Results

Table 2 shows that SAFE-Hair performs best across all five primary metrics. Compared with the strongest prior baseline, SAFE-Hair reduces ChamferPts from 3.74 to 3.28 and improves F-score@4mm/40° from 77.9 to 81.7, indicating more accurate recovery of both geometry and local strand direction. The gain in Invisible-region F-score suggests stronger completion in occluded and back-view regions from a single input image. SAFE-Hair also achieves the highest Silhouette IoU,

Variant	ChamferPts ↓	F-score ↑	F-score (Invisible) ↑	IoU ↑	Penetration Ratio ↓
W/o Scalp UV	3.91	77.8	63.5	0.857	5.1
W/o Rectified Flow	3.67	78.9	64.1	0.866	4.8
W/o Latent Scalp Field	3.58	79.6	66.0	0.871	4.6
W/o Length Factorization	3.49	80.2	67.1	0.876	4.4
W/o Directions	3.43	80.8	68.2	0.879	6.7
Full SAFE-Hair	**3.28**	**81.7**	**69.8**	**0.884**	**3.9**

Table 3: **Ablation study on HairBench-3D.** We evaluate the contribution of the key design choices in SAFE-Hair using the primary benchmark metrics, including geometry accuracy, visible and invisible-region reconstruction quality, silhouette consistency, and physical validity.

Method	RootDist ↓	ValidRoot ↑	FollicleStab. ↑	ExportSucc. ↑
Perm [7]	–	–	–	–
NeuralHDHair [32]	3.8	82.1	–	71.3
Hairstep [40]	2.9	87.4	–	76.8
DiffLocks [26]	2.4	89.6	0.61†	82.5
Im2Haircut [28]	1.7	93.2	0.68†	88.1
Ours	**0.2**	**99.8**	**0.91**	**99.6**

Table 4: **Asset usability on HairBench-3D.** RootDist measures average root-to-scalp distance in mm; ValidRoot and ExportSucc. are percentages. "–" marks metrics that are not applicable, and † indicates approximate follicle stability computed by nearest-root matching. Metric definitions are given in Appendix B.6.

showing better recovery of the global hairstyle outline, and the lowest Penetration Ratio, indicating fewer interpenetration artifacts with the head or body proxy.

In addition to geometric fidelity, we evaluate whether the reconstructed hairstyle behaves as a scalp-rooted strand asset. Table 4 reports root-to-scalp distance, valid-root ratio, follicle stability under small input perturbations, and export success rate. SAFE-Hair achieves high root validity because roots are decoded from fixed canonical follicles rather than predicted as unconstrained 3D points. The active follicle set is also more stable under crop, color, and mask perturbations, which is important for editing and potential temporal use. Table 5 reports side-view inputs: performance degrades gradually with yaw, and at 60° it still matches the strongest baseline's frontal-view results. These results support the central claim that SAFE-Hair reconstructs not only plausible hair geometry but also an exportable groom structure.

Figures 4 and 5 show representative reconstructions exported to Unreal Engine. From a single portrait, SAFE-Hair produces view-consistent strand geometry with plausible back-of-head completion and stable scalp attachment. Compared with prior approaches that often oversmooth occluded regions or produce loose root placement, our results better preserve strand structure, silhouette fidelity, and exportable scalp-rooted geometry. Additional results are shown in Appendix C.

4.5 Ablation Studies

Table 3 analyzes the contribution of each component. Removing scalp-UV conditioning causes the largest degradation, increasing ChamferPts from 3.28 to 3.91 and reducing Invisible-region F-score from 69.8 to 63.5. This shows that the canonical scalp-UV domain is crucial for stable root localization and coherent completion, especially in invisible side and back regions. Without this anchoring, predicted strands become more weakly attached and less spatially organized.

Removing rectified flow also degrades geometry and invisible-region completion, indicating that flow-based distributional refinement helps infer plausible structures beyond the visible input. The latent scalp field improves geometry and silhouette consistency by sharing spatially coherent features across neighboring scalp regions. As shown in Figure 5, removing the latent field preserves the coarse hairstyle shape but weakens local strand coherence and volume consistency.

Input yaw	ChamferPts ↓	F-score ↑	F-score (Invisible) ↑	IoU ↑
0°	3.28	81.7	69.8	0.884
30°	3.35	81.0	68.9	0.878
60°	3.57	79.4	66.2	0.861
90°	3.86	77.2	62.8	0.839

Table 5: **Generalization to side-view inputs on HairBench-3D.** Each held-out hairstyle is rendered at four input yaw angles and reconstructed from the corresponding portrait.

Without length factorization, performance drops for hairstyles with large strand-length variation, leading to worse ChamferPts and Invisible-region F-score. Removing directional supervision has a smaller effect on point-level geometry but increases Penetration Ratio from 3.9 to 6.7, suggesting that tangent-aware prediction is important for physically plausible growth directions near the scalp.

5 Conclusion and Limitations

We introduced SAFE-Hair, a single-view hair reconstruction framework for export-ready strand assets. By reconstructing hair in a canonical scalp-UV domain, generating latent scalp fields with Rectified Flow, and decoding strands from persistent canonical follicles, SAFE-Hair produces rooted strands with stable follicle identities, guide–child organization, and simulation-oriented attributes. Attachment, collision, and bending priors further improve structural validity by reducing off-scalp drift and head interpenetration. We also proposed HairBench-3D, a unified benchmark for evaluating single-view 3D hair reconstruction across strands, meshes, implicit fields, and other hair proxies. With standardized preprocessing, rendering, alignment, and metrics, HairBench-3D enables controlled comparison under identical inputs and evaluation code. Experiments demonstrate that SAFE-Hair improves geometric accuracy, directional consistency, invisible-region completion, render-space agreement, and physical plausibility over representative baselines.

Limitations. SAFE-Hair remains limited by the ambiguity of single-view reconstruction. Severe occlusions, unusual hairstyles, dense accessories, or extreme out-of-distribution grooming patterns can lead to inaccurate invisible-region completion. The current model is designed for static images and does not explicitly enforce temporal consistency for video. In addition, although SAFE-Hair exports simulation-oriented strands, production use may still require solver-specific tuning, material calibration, or artist editing. Future work will explore stronger completion priors, temporal modeling, and closer integration with production grooming and simulation tools.

Acknowledgments and Disclosure of Funding

The work is supported in part by the Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321, ITF PRP/046/24FX, the Science & Technology Cooperation Program of Shandong under grant No. SDST26EG01, SAIL Research Project, and HKUST-Zeekr Collaborative Research Fund.

References

[1] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Transactions on Graphics (TOG)*, 27(3), 2008. doi: 10.1145/1360612.1360662.

[2] Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)*, 31(4):116:1–116:8, 2012. doi: 10.1145/2185520.2185612.

[3] Menglei Chai, Tianjia Shao, Hongzhi Wu, Yanlin Weng, and Kun Zhou. AutoHair: Fully automatic hair modeling from a single image. *ACM Transactions on Graphics (TOG)*, 35(4):116:1–116:12, 2016. doi: 10.1145/2897824.2925961.

[4] Matt Jen-Yuan Chiang, Benedikt Bitterli, Chuck Tappan, and Brent Burley. A practical and controllable hair and fur model for production path tracing. *Computer Graphics Forum*, 35(2):275–283, 2016. doi: 10.1111/cgf.12830.

[5] Gilles Daviet. Interactive hair simulation on the GPU using ADMM. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. doi: 10.1145/3588432.3591509.

[6] Shaocong Dong, Lihe Ding, Xiao Chen, Yaokun Li, Yuxin Wang, Yucheng Wang, Qi Wang, Jaehyeok Kim, Chenjian Gao, Zhanpeng Huang, Zibin Wang, Tianfan Xue, and Dan Xu. From one to more: Contextual part latents for 3d generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8230–8240, 2025.

[7] Chengan He, Xin Sun, Zhixin Shu, Fujun Luan, Soren 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. In *International Conference on Learning Representations (ICLR)*, 2025. Spotlight.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6840–6851, 2020.

[9] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (TOG)*, 34(4):125:1–125:9, 2015.

[10] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. USC-HairSalon: A 3D hairstyle database for hair modeling. `https://huliwenkidkid.github.io/liwenhu.github.io/`, 2015. Public 3D hairstyle database for research and non-commercial use; accessed: 2026-05-05.

[11] Leyang Jin, Yujian Zheng, Bingkui Tong, Yuda Qiu, Zhenyu Xie, and Hao Li. Hairorbit: Multi-view aware 3d hair modeling from single portraits. *arXiv preprint arXiv:2604.02867*, 2026.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

[13] Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. DeepMVSHair: Deep hair modeling from sparse views. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. doi: 10.1145/3550469.3555385.

[14] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[15] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

[16] Pengyu Long, Zijun Zhao, Min Ouyang, Qingcheng Zhao, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. TANGLED: Generating 3D hair strands from images with arbitrary styles and viewpoints. *arXiv preprint arXiv:2502.06392*, 2025.

[17] Haimin Luo, Min Ouyang, Zijun Zhao, Suyi Jiang, Longwen Zhang, Qixuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. GaussianHair: Hair modeling and rendering with light-aware gaussians. *arXiv preprint arXiv:2402.10483*, 2024.

[18] Linjie Luo, Hao Li, Sylvain Paris, Thibaut Weise, Mark Pauly, and Szymon Rusinkiewicz. Multi-view hair capture using orientation fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1490–1497, 2012. doi: 10.1109/CVPR.2012.6247843.

[19] Yue Ma, Yingqing He, Xiaodong Cun, Xintao Wang, Siran Chen, Xiu Li, and Qifeng Chen. Follow your pose: Pose-guided text-to-video generation using pose-free videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4117–4125, 2024.

[20] Yue Ma, Kunyu Feng, Zhongyuan Hu, Xinyu Wang, Yucheng Wang, Mingzhe Zheng, Xuanhua He, Chenyang Zhu, Hongyu Liu, Yingqing He, et al. Controllable video generation: A survey. *arXiv preprint arXiv:2507.16869*, 2025.

[21] Yue Ma, Jiangming Wang, Yucheng Wang, Xilai Wang, Zhiyuan Li, Xinyu Wang, Hongyu Liu, Ruofan Liang, Songchun Zhang, Yuxuan Xue, et al. Livelight: Real-time streaming video relighting with interactive control. *arXiv preprint arXiv:2608.01771*, 2026.

[22] Steve Marschner, Henrik Wann Jensen, Mike Cammarano, Steven Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Transactions on Graphics (TOG)*, 22(3):780–791, 2003. doi: 10.1145/882262.882345.

[23] Giljoo Nam, Chenglei Wu, Min H. Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 155–164, 2019. doi: 10.1109/CVPR.2019.00024.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[25] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands: Learning hair geometry and appearance from multi-view images. In *Computer Vision – ECCV 2022*, pages 73–89. Springer, 2022. doi: 10.1007/978-3-031-19824-3_5.

[26] Radu Alexandru Rosu, Keyu Wu, Yao Feng, Youyi Zheng, and Michael J. Black. DiffLocks: Generating 3D hair from a single image using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10847–10857, 2025.

[27] 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)*, pages 19762–19773, 2023.

[28] Vanessa Sklyarova, Egor Zakharov, Malte Prinzler, Giorgio Becherini, Michael J. Black, and Justus Thies. Im2Haircut: Single-view strand-based hair reconstruction for human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10656–10665, 2025.

[29] Yusuke Takimoto, Hikari Takehara, Hiroyuki Sato, Zihao Zhu, and Bo Zheng. Dr.Hair: Reconstructing scalp-connected hair strands without pre-training via differentiable rendering of line segments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20601–20611, June 2024.

[30] Yucheng Wang and Dan Xu. Modit: Learning highly consistent 3d motion coefficients with diffusion transformer for talking head generation. *arXiv preprint arXiv:2507.05092*, 2025.

[31] Yucheng Wang, Zedong Wang, Yuetong Wu, Yue Ma, and Dan Xu. Care-edit: Condition-aware routing of experts for contextual image editing. *arXiv preprint arXiv:2603.08589*, 2026.

[32] Keyu Wu, Yifan Ye, Lingchen Yang, Hongbo Fu, Kun Zhou, and Youyi Zheng. NeuralHDHair: Automatic high-fidelity hair modeling from a single image using implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, 2022.

[33] Keyu Wu, Lingchen Yang, Zhiyi Kuang, Yao Feng, Xutao Han, Yuefan Shen, Hongbo Fu, Kun Zhou, and Youyi Zheng. MonoHair: High-fidelity hair modeling from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24164–24173, June 2024.

[34] Cem Yuksel. HAIR model files. `https://www.cemyuksel.com/research/hairmodels/`. Public 3D hair model files and HAIR file format specification; accessed: 2026-05-05.

[35] Cem Yuksel, Scott Schaefer, and John Keyser. Hair meshes. *ACM Transactions on Graphics (TOG)*, 28(5): 166:1–166:7, 2009.

[36] Egor Zakharov, Vanessa Sklyarova, Michael J. Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. Human hair reconstruction with strand-aligned 3D gaussians. In *Computer Vision – ECCV 2024*, pages 409–425. Springer, 2024. doi: 10.1007/978-3-031-72640-8_23.

[37] Meng Zhang and Youyi Zheng. Hair-GAN: Recovering 3D hair structure from a single image using generative adversarial networks. *Visual Informatics*, 3(2):102–112, 2019.

[38] Meng Zhang, Menglei Chai, Hongzhi Wu, Hao Yang, and Kun Zhou. A data-driven approach to four-view image-based hair modeling. *ACM Transactions on Graphics (TOG)*, 36(4):156:1–156:11, 2017. doi: 10.1145/3072959.3073664.

[39] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

[40] Yujian Zheng, Zirong Jin, Moran Li, Haibin Huang, Chongyang Ma, Shuguang Cui, and Xiaoguang Han. HairStep: Transfer synthetic to real using strand and depth maps for single-view 3D hair modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12726–12735, 2023.

[41] Yujian Zheng, Yuda Qiu, Leyang Jin, Chongyang Ma, Haibin Huang, Di Zhang, Pengfei Wan, and Xiaoguang Han. Towards unified 3d hair reconstruction from single-view portraits. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[42] Yi Zhou, Liwen Hu, Jun Xing, Weikai Chen, Han-Wei Kung, Xin Tong, and Hao Li. HairNet: Single-view hair reconstruction using convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018.

[43] Yi Zhou, Xin Sun, and Chengan He. Hair20K: A large 3D hairstyle database for hair modeling. `https://zhouyisjtu.github.io/project_hair/hair20k.html`, 2024. 3D strand-based hairstyle dataset containing 21,054 hairstyles; accessed: 2026-05-05.

[44] Yuxiao Zhou, Menglei Chai, Alessandro Pepe, Markus Gross, and Thabo Beeler. GroomGen: A high-quality generative hair model using hierarchical latent representations. *ACM Transactions on Graphics (TOG)*, 42(6):270:1–270:16, 2023. doi: 10.1145/3618309.

[45] Yuxiao Zhou, Menglei Chai, Daoye Wang, Sebastian Winberg, Erroll Wood, Kripasindhu Sarkar, Markus Gross, and Thabo Beeler. GroomCap: High-fidelity prior-free hair capture. *ACM Transactions on Graphics (TOG)*, 43(6):254:1–254:15, 2024. doi: 10.1145/3687768.

Appendix Overview

This appendix provides additional details for SAFE-Hair and HairBench-3D. Sec. A describes implementation details of SAFE-Hair, including canonical scalp-field construction, strand supervision, scalp-conditioned Rectified Flow inference, sampling steps and scalability, training objectives, geometric-validity priors, and export metadata. Sec. B presents the construction and evaluation protocol of HairBench-3D, including data sources, split construction, canonical registration, metric definitions, baseline conversion, asset-usability metrics, and robustness to head-and-camera fitting errors. Sec. C provides additional qualitative results on in-the-wild portrait images and simulation video. Together, these sections are intended to make the method, benchmark, and evaluation protocol easier to reproduce.

A Additional Method Details

A.1 Motivation

We make a distinction between reconstructing hair geometry and reconstructing a usable hair asset. A usable groom is not merely a visually plausible surface or collection of curves: it should contain scalp-attached roots, persistent strand or follicle identities, and a structure that can be serialized, edited, and simulated in downstream pipelines. These requirements are especially important for operations such as guide selection, child-strand interpolation, collision handling, and temporal editing, all of which are naturally defined relative to scalp-rooted strand identities. Existing single-view methods can produce compelling renderings, but their output representations do not always preserve such asset-level structure, particularly in self-occluded regions.

A.2 Canonical scalp-field construction

Each training hairstyle is represented as a set of 3D strands and is first transformed into the coordinate frame of the fitted template head. We use the fitted scalp surface as a shared domain for assigning raw strands to a fixed set of canonical follicles. Let $\mathcal{R} = \{\mathbf{u}_r\}_{r=1}^{R}$ denote canonical follicles in scalp UV space, and let $\psi(\mathbf{u}_r)$ be the corresponding 3D scalp point. For each ground-truth strand $\mathbf{X}_i^\star$, we assign its root to the nearest canonical follicle by scalp geodesic distance:

$$r(i) = \arg\min_r d_{\text{geo}}\left(\mathbf{x}_{i,1}^\star, \psi(\mathbf{u}_r)\right). \tag{12}$$

A follicle is marked as occupied if at least one strand is assigned to it. When multiple strands are assigned to the same follicle, we keep the strand whose root is closest to the canonical follicle location. Each occupied follicle stores an arc-length-resampled strand $\mathbf{X}_r^\star = \{\mathbf{x}_{r,t}^\star\}_{t=1}^{T}$. From this strand we compute the total length $L_r^\star$, normalized segment-length weights $q_{r,t}^\star$, and unit segment directions $\mathbf{d}_{r,t}^\star$. Specifically, $L_r^\star$ is the sum of segment lengths, $q_{r,t}^\star$ is each segment length divided by $L_r^\star + \epsilon$, and $\mathbf{d}_{r,t}^\star$ is the normalized vector from $\mathbf{x}_{r,t}^\star$ to $\mathbf{x}_{r,t+1}^\star$. This decomposition matches the decoder in the main paper: length controls the global extent, segment weights distribute length along the curve, and directions encode local flow.

The per-follicle quantities are then rasterized to the scalp UV grid. Each UV cell stores local statistics of nearby follicles, including occupancy and compressed geometry features. A lightweight encoder maps this rasterized representation to the latent target field z_1 used by the Rectified Flow model. Empty follicles contribute occupancy labels but are ignored by geometry and descriptor losses. Using a latent scalp field, rather than decoding all strands independently, lets neighboring follicles share context and improves completion in weakly observed regions.

A.3 Strand-level supervision

The direct strand supervision term is

$$\mathcal{L}_{\text{strand}} = \lambda_{\text{occ}}\mathcal{L}_{\text{occ}} + \lambda_L\mathcal{L}_L + \lambda_q\mathcal{L}_q + \lambda_d\mathcal{L}_d + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}}. \tag{13}$$

We use binary cross-entropy for occupancy, L_1 losses for total length and segment-length weights, a cosine loss for directions, and an L_1 loss for integrated control points. The occupancy loss is evaluated over all canonical follicles, while geometry losses are evaluated only on occupied follicles.

Occupancy determines which follicles produce active strands and therefore affects the hairline, parting region, and scalp coverage. At inference time, a follicle is kept if its predicted occupancy probability exceeds τ_{occ}. Length and segment-allocation terms supervise complementary aspects of the strand: the former controls coarse extent, while the latter controls how length is distributed along the curve. The direction loss preserves local flow, and the point-level loss anchors the integrated curve in 3D space. In all experiments, the sub-loss weights are selected on the validation split and fixed for final evaluation:

$$\lambda_{\text{occ}} = 1.0, \quad \lambda_L = 0.5, \quad \lambda_q = 0.2, \quad \lambda_d = 1.0, \quad \lambda_{\text{pos}} = 1.0.$$

The HairBench-3D test split is not used for loss-weight selection. Unless otherwise specified, we keep follicles with $o_r > \tau_{\text{occ}}$, where $\tau_{\text{occ}} = 0.5$.

A.4 Scalp conditioning and Rectified Flow inference details

We fit a template head and camera to each input portrait and use the recovered projection to lift image features into the canonical scalp-UV domain. Dense image features are extracted with DINOv2 ViT-B/14 from a 512×512 input crop. For each scalp-UV location, the corresponding 3D scalp point is projected into the image, and the image feature map is bilinearly sampled. Visibility is computed from the fitted head geometry and camera projection.

UV cells that are not directly visible are filled by nearest-neighbor propagation from visible scalp cells and are accompanied by a binary visibility mask. The projected feature map, visibility mask, and positional encodings are concatenated and passed to the conditional Rectified Flow model.

The Rectified Flow network predicts $v_\theta(z_t, t, C)$, where C is the scalp-aligned conditioning map. At inference time, we sample $z_0 \sim \mathcal{N}(0, I)$ and solve the ODE from $t = 0$ to $t = 1$ with 20 Euler steps. Inference therefore uses a fixed number of network evaluations and no per-subject test-time optimization. For evaluation, we use one random sample with a fixed seed, without best-of-K selection. Multiple samples can produce different plausible completions for unobserved side and back regions. Over five seeds, the standard deviation is small for all metrics (Table 2), and variations concentrate in unobserved regions.

A.5 Sampling steps and scalability

Quality largely saturates at 20 Euler steps, which we use by default (Table 6). Increasing the follicle count N or the control points per strand T improves fidelity at roughly linear decoding cost (Table 7); the Rectified Flow cost is unchanged because the latent scalp field stays at 128×128.

Steps	Time (s) ↓	ChamferPts ↓	F-score ↑	F-score (Invisible) ↑
5	1.4	3.47	79.8	67.4
10	2.4	3.34	81.0	69.0
20	4.2	3.28	81.7	69.8
40	7.8	3.26	81.9	70.0

Table 6: **Effect of Rectified Flow sampling steps on HairBench-3D.**

Follicles N	Points T	Time (s) ↓	Memory (GB) ↓	ChamferPts ↓	F-score ↑
32,768	32	4.2	7.9	3.28	81.7
65,536	32	6.8	12.7	3.22	82.1
32,768	64	5.6	10.8	3.21	82.0
65,536	64	9.4	18.6	3.17	82.5

Table 7: **Scalability in follicle count N and control points per strand T.**

A.6 Training objectives and geometric-validity priors

For real portraits without 3D strand ground truth, we supervise the prediction by differentiable rendering. The predicted strands are rendered from the estimated input camera and compared with

image-derived hair cues. The silhouette term compares the rendered hair mask with the observed or estimated hair mask, constraining the visible hair volume and outer contour. The orientation term compares the rendered 2D strand orientation field with an image-derived orientation field over visible hair pixels, providing local flow supervision. Since a single portrait observes only part of the hair, these image-space losses are complemented by the conditional scalp-field prior, which learns plausible completions from 3D hair data.

The geometric-validity loss is

$$\mathcal{L}_{\text{valid}} = \lambda_{\text{att}}\mathcal{L}_{\text{att}} + \lambda_{\text{col}}\mathcal{L}_{\text{col}} + \lambda_{\text{bend}}\mathcal{L}_{\text{bend}}. \tag{14}$$

These terms reduce common asset-level failures but are not intended to replace physical simulation. They are evaluated on decoded strands during training and require no iterative correction at test time.

The attachment prior encourages the first strand segment to leave the scalp in a plausible direction:

$$\mathcal{L}_{\text{att}} = \frac{1}{|\mathcal{R}^+|} \sum_{r\in\mathcal{R}^+} \max\left(0, \gamma_{\text{att}} - \mathbf{n}_r^\top \hat{\mathbf{d}}_{r,1}\right), \tag{15}$$

where $\mathbf{n}_r$ is the outward scalp normal and $\mathcal{R}^+$ is the set of occupied follicles. The collision prior penalizes decoded samples that enter the fitted head mesh or lie within a small safety margin:

$$\mathcal{L}_{\text{col}} = \frac{1}{|\hat{\mathcal{X}}|} \sum_{\hat{\mathbf{x}}\in\hat{\mathcal{X}}} \max\left(0, \delta_{\text{col}} - s_M(\hat{\mathbf{x}})\right), \tag{16}$$

where s_M is the signed distance to the fitted head mesh. The bending prior regularizes high-frequency direction changes:

$$\mathcal{L}_{\text{bend}} = \frac{1}{|\mathcal{R}^+|(T-2)} \sum_{r\in\mathcal{R}^+} \sum_{t=1}^{T-2} \left(1 - \hat{\mathbf{d}}_{r,t}^\top \hat{\mathbf{d}}_{r,t+1}\right). \tag{17}$$

We use

$$\lambda_{\text{att}} = 0.1, \quad \lambda_{\text{col}} = 0.5, \quad \lambda_{\text{bend}} = 0.05, \quad \gamma_{\text{att}} = 0.15, \quad \delta_{\text{col}} = 2\,\text{mm}.$$

These values are selected on the validation split and fixed for all experiments.

A.7 Export metadata and guide-family construction

The decoder may optionally predict a low-dimensional descriptor $\mathbf{e}_r$ for each active follicle. This descriptor is used as export metadata rather than as the primary reconstruction target. Depending on available supervision, it may include geometry-derived quantities such as total length, mean direction, curvature statistics, curl, or other groom-control channels. When no descriptor targets are available, the descriptor loss is disabled. For descriptor supervision, we use uniform descriptor weights $w_m = 1$ and global weight $\lambda_{\text{desc}} = 0.01$.

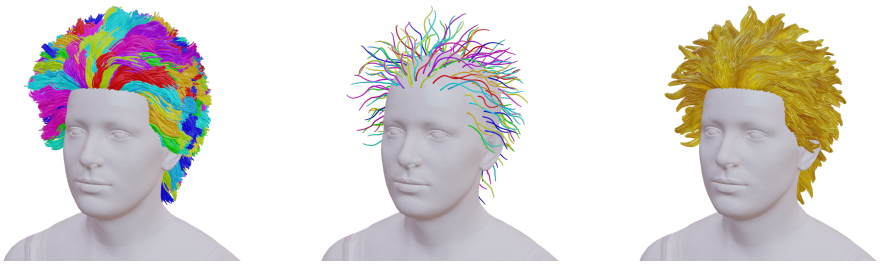
When curl supervision is used, we define curl as the average change of tangent direction along the resampled strand:

$$c_r^\star = \frac{1}{T-2} \sum_{t=1}^{T-2} \arccos\left(\text{clip}\left({\mathbf{d}_{r,t}^\star}^\top \mathbf{d}_{r,t+1}^\star, -1, 1\right)\right). \tag{18}$$

Descriptor supervision is applied only to occupied follicles and only to dimensions with available targets:

$$\mathcal{L}_{\text{desc}} = \frac{1}{|\mathcal{R}^+|} \sum_{r\in\mathcal{R}^+} \sum_{m\in\mathcal{M}_r} w_m \left|e_{r,m} - e_{r,m}^\star\right|. \tag{19}$$

Guide-family construction is performed after strand reconstruction and does not affect predicted geometry. We select guide strands from active follicles using farthest-point sampling in scalp UV space. Optionally, the sampling score can be weighted by strand length or curvature so that visually important strands are more likely to become guides. Each remaining strand is assigned to the nearest guide using the joint scalp-position and strand-feature distance from the main paper. The exported asset stores control points, follicle ID, scalp UV coordinate, optional descriptor, and optional guide ID, which supports curve-based groom serialization and downstream editing.



(a) Cluster (b) Guide strands (c) Refinement

Figure 6: **Exportable groom structure.** SAFE-Hair groups active follicles into guide families in canonical scalp UV space. Each child strand stores a persistent follicle ID, guide ID, control points, and optional physical or groom-control attributes.

B HairBench-3D Benchmark Details

B.1 Data Sources and Licenses

HairBench-3D is constructed from strand-level hairstyle assets collected from public or research-available sources, including Hair20K, USC-HairSalon, and Yuksel Hair Meshes [10, 35, 43]. We use only assets that can be converted to explicit 3D strands with root positions, ordered control points, and strand tangents. Assets with missing geometry, ambiguous scale, corrupted strand order, or severe head misalignment are excluded.

For each source, we record the original asset identifier, source dataset, license or usage terms, and processed split assignment. The benchmark release contains only assets or derived annotations permitted by the source terms. When raw asset redistribution is not permitted, we provide preprocessing scripts, split files, and instructions for obtaining the original data from the official source. Because different sources may contain shared or near-identical hairstyles, we perform duplicate and near-duplicate detection before assigning train, validation, or test membership. The official test split is frozen and is not modified after model development begins.

B.2 Split Construction and Leakage Prevention

We construct the split at the asset level rather than the rendered-image level. All rendered views, augmentations, and processed variants of the same hairstyle inherit the same split assignment. This prevents different views of one hairstyle from appearing in both training and test sets.

We compute asset descriptors for duplicate detection, including scalp-root distributions, strand-length histograms, root-direction histograms, rendered silhouettes from canonical views, and global geometric statistics. Potential near-duplicates are identified using these descriptors and are manually inspected or conservatively assigned to the same split.

The final HairBench-3D test split contains 1,551 hairstyles. No test asset, rendered input, augmented variant, near-duplicate, or processed derivative is included in the SAFE-Hair training or validation pool. The test split is not used for hyperparameter selection, loss-weight tuning, filtering-threshold selection, early stopping, model selection, or qualitative example selection.

B.3 Canonical Registration and Scalp-UV Parameterization

All benchmark assets are converted to a shared canonical coordinate system using a template head with a predefined scalp mask and scalp-UV parameterization. For each hairstyle, we estimate a rigid alignment to the template head using the associated head mesh or verified landmark correspondences. Each strand root is then projected onto the closest point on the template scalp.

A strand is considered valid if its projected root lies inside the scalp mask and its root-to-scalp distance is below a fixed threshold. Invalid strands are discarded. Each valid strand is assigned a canonical follicle index by nearest-neighbor lookup on the scalp surface and resampled to T ordered control points by arc length. We store the resampled points, unit tangents, total length, root location,

Usage of HairBench-3D test split	Allowed?
Training data	No
Validation or hyperparameter tuning	No
Loss-weight selection	No
Filtering-threshold selection	No
Qualitative example selection before final evaluation	No
Final quantitative evaluation	Yes
Final qualitative visualization of evaluated samples	Yes

Table 8: Usage policy for the frozen HairBench-3D test split.

follicle index, and visibility labels. This canonicalization ensures that all methods are evaluated under the same scale, orientation, and head geometry. It also enables root-aware, direction-aware, and invisible-region metrics across methods with different native representations.

B.4 Metric Definitions

Let $P = \{(p_i, d_i)\}$ denote predicted strand samples and $G = \{(g_j, t_j)\}$ denote ground-truth samples. All points are expressed in the canonical coordinate system and distances are measured in millimeters.

ChamferPts. ChamferPts measures symmetric point-level distance:

$$\mathrm{ChamferPts}(P, G) = \frac{1}{|P|} \sum_{p_i \in P} \min_{g_j \in G} \|p_i - g_j\|_2 + \frac{1}{|G|} \sum_{g_j \in G} \min_{p_i \in P} \|g_j - p_i\|_2. \tag{20}$$

We uniformly sample strand points by arc length before computing the metric.

F-score@4 mm/40°. A predicted sample is matched if there exists a ground-truth sample satisfying

$$\|p_i - g_j\|_2 < 4\,\mathrm{mm} \quad \text{and} \quad \arccos(|d_i^\top t_j|) < 40^\circ. \tag{21}$$

Precision and recall are computed from matched predicted and ground-truth samples, and the F-score is

$$F = \frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall} + \epsilon}. \tag{22}$$

Invisible-region F-score. Invisible-region F-score uses the same distance and tangent thresholds as above but restricts ground-truth samples to those invisible from the input view. Visibility labels are computed using z-buffer visibility on rendered ground-truth strand samples. This metric evaluates completion in self-occluded side and back regions.

Silhouette IoU. We render predicted and ground-truth hairstyles from held-out views using the same camera and head geometry. Silhouette IoU is

$$\mathrm{IoU} = \frac{|M_{\mathrm{pred}} \cap M_{\mathrm{gt}}|}{|M_{\mathrm{pred}} \cup M_{\mathrm{gt}}|}. \tag{23}$$

Penetration Ratio. Penetration Ratio measures the fraction of predicted samples inside the head or body proxy:

$$\mathrm{Penetration} = \frac{1}{|P|} \sum_{p_i \in P} \mathbf{1}\left[\mathrm{SDF}_{\mathrm{head/body}}(p_i) < 0\right]. \tag{24}$$

Lower values indicate fewer interpenetration artifacts.

B.5 Baseline Output Conversion

HairBench-3D supports methods with different output representations. For fair evaluation, each output is converted to the representation required by each metric using fixed, method-agnostic rules.

For strand-based methods, we resample predicted strands by arc length and compute tangents from adjacent control points. For mesh-based methods, we uniformly sample surface points and estimate local directions from available strand directions, anisotropic texture orientation, or local principal directions. For implicit or volumetric methods, we extract the hair surface or occupancy boundary using the authors' recommended procedure and sample points from the extracted geometry. For Gaussian- or particle-based representations, we use Gaussian centers or particle locations as point samples and estimate directions if available.

For direction-aware metrics, methods without reliable tangents are evaluated using their best available local orientation estimate. If no reliable orientation is available, we report distance-only metrics in the supplement and mark direction-aware metrics as not directly applicable. In the main comparison, we include only baselines whose outputs can be converted to the required metric inputs without using ground-truth 3D hair. All conversion code and sampling densities are fixed across methods.

Because prior methods are trained on different private, synthetic, or partially released collections, HairBench-3D should be interpreted as a fixed held-out system-level evaluation rather than a claim of identical training data. When trainable code is available, we additionally evaluate retrained or fine-tuned variants under the same training split; otherwise, we use official released weights and document this choice. HairOrbit [11] and TANGLED [16] have released their code for fair comparison.

B.6 Additional asset-usability metrics

HairBench-3D also supports asset-usability metrics that evaluate whether a reconstruction behaves as a scalp-rooted strand groom. Results are reported in Table 4. ExportSucc. is the percentage of predictions that can be serialized as valid strand-groom assets, and "–" marks metrics that a method's output cannot provide. For DiffLocks and Im2Haircut, follicle stability is approximated by matching each predicted root to its nearest canonical follicle. These methods can also be exported after post-processing, whereas SAFE-Hair provides scalp attachment and follicle identities natively.

Root-to-scalp distance. For each active predicted strand, we compute the distance from its root to the canonical scalp surface:

$$d_{\mathrm{root}} = \frac{1}{|\hat{\mathcal{R}}^+|} \sum_{r \in \hat{\mathcal{R}}^+} \mathrm{dist}\left(\hat{\mathbf{x}}_{r,1}, M_{\mathrm{scalp}}\right). \tag{25}$$

Valid-root ratio. We report the fraction of active strands whose roots lie within the scalp mask and within a distance threshold τ_{root}:

$$\mathrm{ValidRoot} = \frac{1}{|\hat{\mathcal{R}}^+|} \sum_{r \in \hat{\mathcal{R}}^+} \mathbf{1}\left[\hat{\mathbf{x}}_{r,1} \in M_{\mathrm{scalp}} \ \wedge \ \mathrm{dist}(\hat{\mathbf{x}}_{r,1}, M_{\mathrm{scalp}}) < \tau_{\mathrm{root}}\right]. \tag{26}$$

We use $\tau_{\mathrm{root}} = 2\,\mathrm{mm}$ in all experiments.

Follicle stability under input perturbations. To measure follicle-identity stability, we evaluate the same image under small crop, color, or segmentation perturbations. Given active follicle sets $\hat{\mathcal{R}}_a^+$ and $\hat{\mathcal{R}}_b^+$, we compute

$$\mathrm{FollicleIoU} = \frac{|\hat{\mathcal{R}}_a^+ \cap \hat{\mathcal{R}}_b^+|}{|\hat{\mathcal{R}}_a^+ \cup \hat{\mathcal{R}}_b^+|}. \tag{27}$$

Guide-family compactness. For exported guide families, we measure the average child-to-guide distance in the joint scalp-position and strand-feature space:

$$\mathrm{GuideCompact} = \frac{1}{|\hat{\mathcal{R}}^+|} \sum_{r \in \hat{\mathcal{R}}^+} \left(\alpha \|\mathbf{u}_r - \mathbf{u}_{g(r)}\|_2^2 + \beta \|\mathbf{f}_r - \mathbf{f}_{g(r)}\|_2^2\right). \tag{28}$$

B.7 Robustness to head-and-camera fitting errors

We perturb the estimated camera by yaw, pitch, and roll offsets sampled from $\mathcal{U}(-5^\circ, 5^\circ)$, and the estimated head shape by Gaussian noise ($0.5\times$ the training-set standard deviation) on the first ten

PCA coefficients; "Combined" applies both. The same perturbations are applied to DiffLocks and Im2Haircut, and all methods are evaluated against the unperturbed ground truth, where RootDist-to-GT measures root distance to the ground-truth scalp. SAFE-Hair degrades the least under every perturbation, most clearly in the root metrics (Tables 9 and 10).

Perturbation	ChamferPts ↓	F-score ↑	RootDist-to-GT ↓	ValidRoot ↑
None	3.28	81.7	0.20	99.8
Camera rotation ($\pm 5^\circ$)	3.33	81.2	0.22	99.6
Head-shape deviation	3.53	79.5	0.96	97.8
Combined	3.62	78.9	1.15	96.7

Table 9: **SAFE-Hair under head-and-camera fitting errors.**

	Camera rotation				Head-shape deviation				Combined			
Method	ΔCD	ΔF	ΔRoot	ΔValid	ΔCD	ΔF	ΔRoot	ΔValid	ΔCD	ΔF	ΔRoot	ΔValid
DiffLocks [26]	2.2%	0.8	0.08	0.8	9.8%	3.1	1.25	4.8	14.2%	4.1	1.52	6.1
Im2Haircut [28]	2.0%	0.7	0.05	0.5	9.4%	2.8	1.02	3.8	12.6%	3.7	1.24	5.0
Ours	**1.5%**	**0.5**	**0.02**	**0.2**	**7.6%**	**2.2**	**0.76**	**2.0**	**10.4%**	**2.8**	**0.95**	**3.1**

Table 10: **Robustness to head-and-camera fitting errors.** Degradation from each method's clean setting (lower is better); ΔF and ΔValid in points, ΔRoot in mm.

C Additional Qualitative Results and Animation Video

We provide additional qualitative results and an accompanying animation video as supplementary.

The qualitative examples show SAFE-Hair reconstructions from in-the-wild portrait images with diverse hairstyles, identities, lighting conditions, and backgrounds. For each example, we visualize the input portrait and the reconstructed 3D hair rendered on a neutral template head. Results are shown in Figures 8 and 9.

The supplementary animation video further shows reconstructed hair assets under novel views and simple motion. It illustrates that SAFE-Hair outputs explicit strand-based assets that can be rendered from different viewpoints and used in downstream animation or editing pipelines. Since the input is a single portrait image, the video demonstrates plausible asset behavior rather than ground-truth temporal reconstruction.

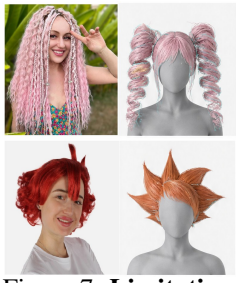


Figure 7: **Limitation.** Extreme Hairstyles



Figure 8: Additional qualitative comparison on in-the-wild portrait images. From left to right: input image, SAFE-Hair, Hairstep, and NeuralHDHair. SAFE-Hair produces more coherent and complete strand reconstructions across diverse hairstyles, including short, straight, wavy, spiky, curly, and afro hair. Compared with baselines, our method better preserves the global hair silhouette and local strand flow, while reducing floating strands, sparse regions, and unstable scalp attachment.



Figure 9: **Additional qualitative results on in-the-wild portrait images.** For each pair, the left image shows the input portrait and the right image shows the reconstructed 3D hair rendered on a neutral template head. These examples complement the supplementary animation video by showing the diversity of single-image reconstructions produced by SAFE-Hair.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The claims are stated in the abstract and introduction and are supported by the method, experiments, and ablations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The paper discusses limitations and failure cases in the limitations section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

Justification: The paper does not present new theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper and appendix describe the model, training setup, data processing, metrics, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes].

Justification: Code is released at the link given in the abstract, and HairBench-3D will be released together with the metric scripts (Section 4).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Training details, data splits, hyperparameters, and evaluation settings are specified in the paper and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Table 2 reports the mean and standard deviation (1-sigma) of all main metrics over five Rectified Flow sampling seeds with fixed model weights; the variability captured is that of the initial noise sample.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The paper reports the hardware, memory, runtime, and training cost used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We reviewed and followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The paper discusses beneficial uses, possible misuse, and mitigation measures.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes].

    Justification: The paper describes license-aware release, consent-based use, and restrictions against deceptive or non-consensual use.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: Existing datasets, assets, code, and models are cited with license information where available.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The new benchmark and released artifacts are documented in the paper, appendix, and supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human-subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human-subject studies, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are not used as a core methodological component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.