HAIRFLOW: MODELING 3D HAIR STRANDS WITH MULTI-MODAL CONDITIONED DIFFUSION

Anonymous authors
Paper under double-blind review

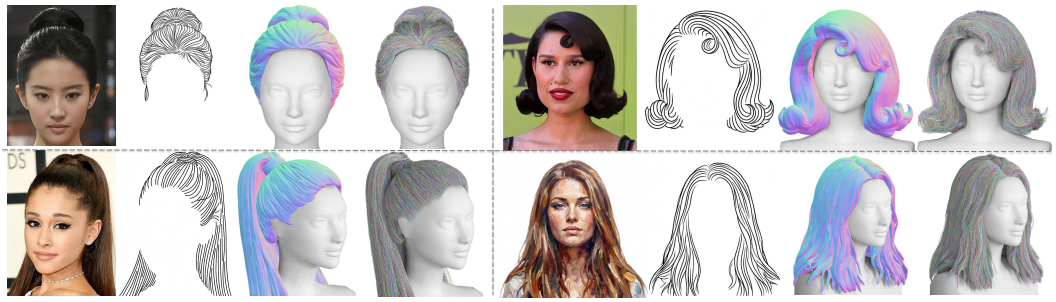

Figure 1: HairFlow uses conditioned diffusion to recover 3D strands from image inputs, guided by 3D mesh and sketches inferred by foundation models. It handles complex hairstyles and stylized images while closely adhering to the input conditions.

ABSTRACT

Modeling 3D hair strands is challenging due to complex hairstyle geometry, scarce high-quality training data, and incomplete or noisy input observations. Foundation models offer broader hairstyle priors, but their outputs are not directly usable for strand-based rendering and simulation. Existing approaches require costly per-instance optimization to fit hair priors to meshes from foundation models and 2D orientations from specialized predictors trained on limited hair data, restricting efficiency and generalization. We present HairFlow, a multimodal conditioned diffusion framework for efficient and controllable hair strands modeling. HairFlow conditions strand generation on two complementary representations: a coarse mesh that constrains global shape and multi-view sketches that specify local strand flow. HairFlow combines an Orientation VAE with a conditioned diffusion transformer to generate volumetric fields that are traced into explicit strands. To overcome the lack of paired sketch–strand training data, we introduce an automatic synthesis pipeline that generates geometrically aligned multi-view sketches with diverse styles representative of inference-time inputs. Experiments show that HairFlow generates plausible 3D strands that faithfully follow the input conditions across diverse hairstyles, substantially reduces inference time, and supports flexible multimodal control.

1 INTRODUCTION

3D hair modeling is essential for digital avatars, games, and visual effects. Strand-based representations are standard in high-end production, enabling high-quality rendering and physically based animation. A hairstyle can contain over 100,000 strands with complex structures such as curls, locks, and braids, making manual authoring through guide placement and grooming highly labor-intensive.

Recent works Zheng et al. (2023); Zhou et al. (2024); Rosu et al. (2022); Kuang et al. (2022); Sklyarova et al. (2025); Rosu et al. (2025) have explored reconstructing strand-level geometry from single- or multi-view images. However, hair reconstruction is inherently underconstrained: images

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

provide incomplete observations of the underlying 3D strand structure, particularly in occluded regions and beneath overlying hair. Existing learning-based methods therefore rely heavily on hairstyle priors learned from 3D hair datasets, whose limited scale and diversity restrict generalization to complex, in-the-wild hairstyles. Foundation models offer an alternative by providing rich geometric and visual priors learned at scale. Large 3D reconstruction models (LRMs), for example, can recover coarse hair geometry, while image generation models can provide visual cues about local hair flow. Recent methods have begun leveraging these representations for strand reconstruction Shen et al. (2026); Hao et al. (2026). HairLRM Shen et al. (2026) fits a learned hair prior to a reconstructed hair mesh and 2D orientation maps to obtain a volumetric orientation field for strand extraction. However, its per-instance latent optimization is costly and limits representation capacity, while its orientation estimator, trained on limited hair data, generalizes poorly to diverse hairstyles.

We present HairFlow, a multimodal conditioned diffusion framework for 3D hair strand modeling. HairFlow uses a coarse hair mesh and multi-view sketches as a unified interface: the mesh constrains global shape and spatial extent, while the sketches specify local strand flow. These conditions can be generated from images or text using foundation models or provided directly by artists, enabling a flexible interface for strand modeling from diverse inputs. At its core, HairFlow learns a compact latent representation of volumetric hair orientations using an Orientation VAE and generates these latents with a conditional diffusion transformer trained via flow matching. Mesh- and camera-aware sketch features are incorporated through cross-attention, enabling direct generation without per-instance optimization. We then decode the generated orientation field into explicit 3D strands. To address the scarcity of paired training data, we also introduce an automatic pipeline to synthesize geometrically aligned multi-view sketches with diverse stroke styles from existing 3D strand assets.

Qualitative comparisons on real-world images and quantitative evaluations on synthetic image–strand pairs demonstrate that HairFlow generates 3D strand geometry more closely aligned with the inputs while reducing end-to-end inference time. We further demonstrate its support for multi-view sketches, diverse sketch and image styles, and artist-provided guidance, as well as the downstream usability of the generated strands in physical simulation.

2 Related Work

Hair Representations Mesh-based (Yuksel et al., 2009) and card-based representations (Zheng et al., 2025; Tojo et al., 2025) are compact and support efficient real-time rendering (Wu & Yuksel, 2016; Huang et al., 2025), but their low-polygon nature precludes accurate simulation. NeRF (Mildenhall et al., 2021) and Gaussian splatting (Kerbl et al., 2023) capture hair appearance effectively for view synthesis (Wang et al., 2025a;b; Zakharov et al., 2024; Wang et al., 2026), yet are unsuitable for physically based rendering (Marschner et al., 2003). Strand-based representations therefore remain the standard for high-end film and game production. Since raw strands are high-dimensional and unstructured, most methods adopt more structured intermediate representations: 3D orientation fields supporting strand growing (Shen et al., 2026; 2023; Wu et al., 2022), scalp-UV parameterizations (Rosu et al., 2025; Zhou et al., 2023), sparse guide hairs for interpolation (Hsu et al., 2024). For image-based strand modeling, 2D geometric cues bridge image appearance and 3D hair representations. Chai et al. (2012) extract orientation maps using Gabor filters, while Zheng et al. (2023) learn to predict strand direction and depth maps. However, filtering is sensitive to image noise and illumination, while learned predictors rely on limited synthetic hair data. We instead use multi-view sparse sketches as higher-level descriptions of hair flow, paired with a coarse mesh specifying the hair envelope, to condition a volumetric orientation field for strand generation.

Hair Strand Modeling High-fidelity strand reconstruction can be achieved using multi-view imagery (Nam et al., 2019; Zhou et al., 2024; Rosu et al., 2022; Kuang et al., 2022) or CT capture (Shen et al., 2023), but requires specialized acquisition setups. Single-image reconstruction Sklyarova et al. (2025); Rosu et al. (2025) offers a more accessible alternative, yet a single view reveals only part of a hairstyle, leaving its depth and occluded structures ambiguous. Early approaches Hu et al. (2014) address this ambiguity by retrieving similar hairstyles from strand assets database. Learning-based methods instead construct and augment 3D strand datasets to learn priors over hair geometry (Rosu et al., 2025; He et al., 2025a; Zheng et al., 2024). Different architectures have been explored: HairStep (Zheng et al., 2023) uses predicted 2D orientation maps as an intermediate representation for recovering 3D hair geometry; DiffLocks (Rosu et al., 2025) formulates strand generation

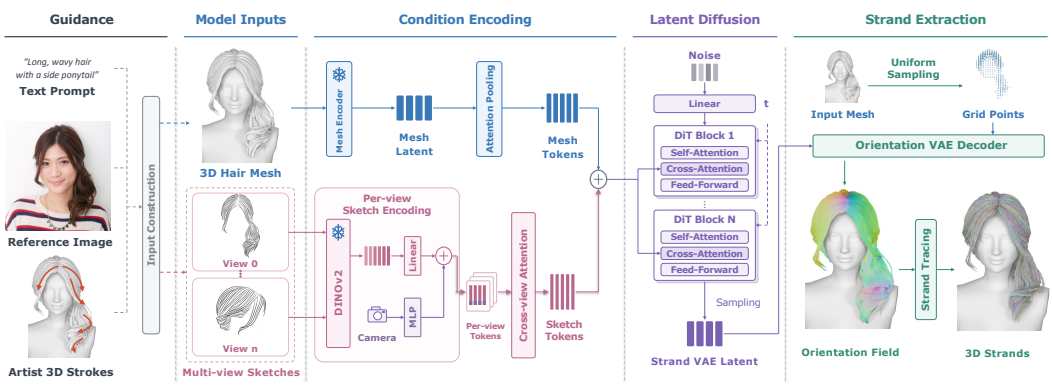

Figure 2: Overview of HairFlow. HairFlow takes two geometric inputs—a coarse hair mesh and multi-view sketches—to specify global shape and local strand flow. These conditions can be obtained from images or text using foundation models, or provided by artists. Mesh and camera-aware sketch encoders produce tokens that condition a diffusion transformer through cross-attention. The transformer generates latents in the space of a pretrained Orientation VAE, which decodes them into an orientation field for explicit strand extraction.

through diffusion; and HairGPT (Luo et al., 2026) and CHARM (He et al., 2025b) adopt autoregressive modeling. Despite these advances, the limited coverage of training data remains a bottleneck for generalization to diverse real-world hairstyles.

Recent work leverages large pretrained models to provide additional geometry evidence for sparse view reconstruction, for example, HairOrbit (Jin et al., 2026) uses a video generation model to synthesize multiple views. Our method follows this line of work and is most closely related to HairLRM (Shen et al., 2026), which takes a 3D mesh from large reconstruction model and a 2D orientation map predicted by HairStep (Zheng et al., 2023) as input. HairLRM jointly represents the hair shell as a signed distance field and the internal strand flow as a volumetric orientation field in one VAE latent space. At inference, it optimizes the latent code to match the decoded distance field to the mesh and the decoded orientations to predicted 2D orientation maps lifted onto the hair mesh, then extracts strands from the completed field. This per-instance optimization is costly, while its flow guidance remains dependent on a predictor trained on limited hair data. A more recent work (Hao et al., 2026) uses a training-free pipeline to convert multi-view renderings of an hair mesh into dense orientation maps via an image-editing model, back-project them onto the mesh, and propagate surface orientations inward using hand-crafted rules. This approach relies on slow iterative propagation, and cues derived from mesh renderings may deviate from the input image.

Beyond real images, hair can be generated from text (Sklyarova et al., 2024; Sun et al., 2025), sketches (Shen et al., 2020; Zhang et al., 2025; Long et al., 2025), or several modalities jointly (Luo et al., 2026; Wei et al., 2022), and recent work targets stylized portraits (He et al., 2025b).

3 Method

Given a hair mesh and sketches, HairFlow generates 3D hair strands that follow the mesh's global shape and the sketches' local flow (Fig. 2). We represent hair strands by volumetric orientation fields that specifies local strand directions, which can be extracted to explicit strands. We learn a compact latent space with an Orientation VAE (Sec. 3.1) and generate latents using a conditional diffusion transformer (Sec. 3.3). Mesh and camera-aware sketch tokens (Sec. 3.2) guide generation through cross-attention. The generated latents are decoded into an orientation field and traced into explicit strands. Sec. 3.4 details data synthesis, training and inference process.

3.1 Orientation VAE

We build our Orientation VAE on the 3DShape2VecSet-based architecture (Zhang et al., 2023) adopted by HairLRM (Shen et al., 2026). HairLRM jointly models a hair SDF and a volumetric

orientation field, optimizing latent codes per instance to fit the input mesh and 2D orientations predicted by HairStep. In contrast, we generate latent codes directly through multimodal conditional diffusion, without per-instance fitting. Our VAE therefore models only the orientation field, omitting the SDF required for mesh fitting in HairLRM and dedicating more model capacity to local strand directions. More importantly, HairLRM employs a compact 512×2 latent representation to keep per-instance optimization tractable, whereas our generative formulation removes this constraint. We therefore increase the latent capacity to 2048×64, enabling finer-grained 3D hair orientation.

Given the latent tensor $\mathbf{z}$ and a query point $\mathbf{x} \in \mathbb{R}^3$, the decoder predicts the local hair 3D orientation

$$\mathbf{o}(\mathbf{x}) = D_{\text{hair}}(\mathbf{z}, \mathbf{x}), \qquad \mathbf{o}(\mathbf{x}) \in \mathbb{R}^3. \tag{1}$$

Because the diffusion model operates directly in the strand latent space, only the decoder is required at inference. Given a sampled latent $\mathbf{z}$ and the coarse hair mesh M, we uniformly sample 3D query points $X = \{\mathbf{x}_i\}_{i=1}^{N}$ inside M and evaluate D_{hair} at these points to obtain a dense volumetric orientation field, from which explicit hair strands can be extracted.

3.2 Multi-Modal Conditioning

Our method adopts a unified condition interface consisting of a coarse hair mesh and multi-view sketches. The mesh constrains hair length, volume, and silhouette, while the sketches specify local strand directions and finer styling details. These modalities differ substantially: the mesh provides dense 3D geometry without directions, whereas sketches convey sparse directional cues in 2D projections. Other modalities, such as images, text, and videos, can also be supported by converting them into these two basic inputs. Our conditioning module aligns these complementary signals to guide diffusion toward hair structures consistent with both.

Mesh conditioning To obtain a compact and structured conditioning signal for the diffusion model, we encode the input mesh M with a pretrained 3D mesh VAE encoder (Chen et al., 2025; Zhang et al., 2023) and further compress the resulting latent tokens with a learned attention pooling module:

$$\tilde{c}_{\text{mesh}} = P_{\text{mesh}}\big(E_{\text{mesh}}(M)\big), \tag{2}$$

where $E_{\text{mesh}}(M) \in \mathbb{R}^{N_m \times d_m}$ is the mesh latent produced by the frozen encoder and $\tilde{c}_{\text{mesh}} \in \mathbb{R}^{N'_m \times d}$ is the pooled mesh condition consumed by the transformer. Here N_m and N'_m are the numbers of tokens before and after pooling, d_m is the mesh latent dimension, and d is the transformer hidden dimension. Rather than treating all mesh tokens equally, the pooling module learns to weight and aggregate mesh geometric features according to their relevance to the hair generation process.

Sketch conditioning Input sketches may come from general-purpose image-editing models or artists, resulting in diverse stroke styles and densities. We therefore seek features that capture local hair directions while remaining robust to appearance variations, and adopt DINOv2 (Oquab et al., 2023) as our feature backbone. We observe that shallow features are overly sensitive to stroke density and width, whereas deep features emphasize shared semantic content, treating patches with different hair directions similarly. We therefore concatenate features from several intermediate layers to balance directional sensitivity and appearance robustness. Given V sketches S^v and their corresponding orthographic cameras C^v, we extract these features independently for each view, incorporate camera information, and fuse the resulting tokens across views using self-attention.

To inject camera parameters, we compute a 6D Plücker ray $r_i^v \in \mathbb{R}^6$ for the center of each patch under camera C^v. Plücker parameterization encodes a ray directly rather than a camera pose, so it is independent of any global coordinate convention and varies smoothly with the viewpoint. We apply separate learned projections to the DINOv2 feature and the Plücker ray encoding, and concatenate the resulting embeddings to form a camera-aware patch token:

$$t_i^v = [\phi_f(f_i^v);\ \phi_r(r_i^v)], \tag{3}$$

where ϕ_f and ϕ_r are learned projection layers. We then flatten the tokens from all views and process them jointly with a transformer to enable cross-view reasoning:

$$c_{\text{sketch}} = F_{\text{sketch}}(\mathcal{T}_{\text{sketch}}), \tag{4}$$

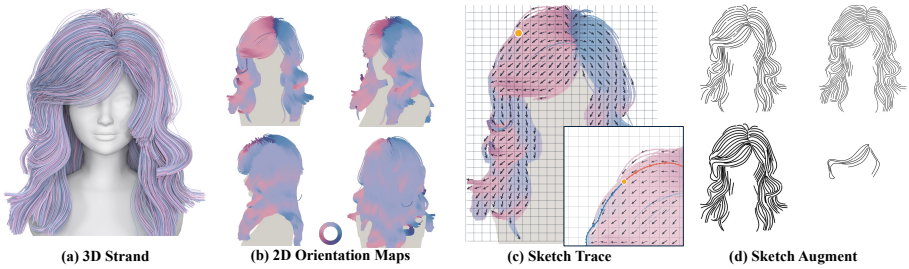

Figure 3: Hair sketch synthesis pipeline. (a) Input strands and template head mesh. (b) Multi-view orientation maps; colors encode directed 2D tangents as shown in the circular legend. (c) Bidirectional tracing from sampled seeds; arrows show local directions, and the inset details tracing from the highlighted seed. (d) Sketch variations through stroke density, line width, and localized seed sampling.

where $\mathcal{T}_{\text{sketch}} = \{t_i^v\}_{v=1,\,i=1}^{V,\,N_p}$ is the flattened token sequence over all views N_p is the number of patches per view and $c_{\text{sketch}} \in \mathbb{R}^{(VN_p)\times d}$ is the final sketch conditioning. Cross-attention over c_{sketch} naturally accommodates a variable number of input views.

3.3 Strand Diffusion Transformer

We use a conditional diffusion transformer to generate hairstyles in the orientation VAE latent space. Mesh and sketch information are provided through a joint conditioning sequence:

$$\mathbf{c} = [\tilde{c}_{\text{mesh}};\ c_{\text{sketch}}], \tag{5}$$

where concatenation is along the token dimension. Given noisy latent tokens $\mathbf{z}_t \in \mathbb{R}^{L\times d_z}$, the transformer integrates these conditions through cross-attention, with the time variable t incorporated via adaLN modulation. The network predicts a latent velocity field $\hat{\mathbf{v}} \in \mathbb{R}^{L\times d_z}$ and is trained using the flow-matching objective described in Sec. 3.4.

3.4 Training and Inference

Training Data We collect approximately 55K strand-based hairstyles from four publicly available datasets: 40K from DiffLocks (Rosu et al., 2025), 10K from Perm (He et al., 2025a), 2K from UniHair (Zheng et al., 2024), and 3K from HairCS Lu et al. (2026). All hairstyles are aligned to a common coordinate frame. Following HairLRM (Shen et al., 2026), we extract a hair surface mesh and orientation fields from the ground-truth 3D strands. We also develop a pipeline to synthesize 2D sketches of strands from multiple viewpoints and in diverse styles, as detailed in next ection.

Sketch Synthesis At inference time, sketches can be generated by image-editing models or drawn by artists, but collecting them at scale with accurate 3D strand alignment is challenging. Converting strand renderings into sketches with image-editing models is also too time-consuming at scale. We therefore develop an efficient pipeline to synthesize aligned, diverse training sketches directly from 3D strands(Fig. 3).

We first convert 3D strands into a 3D orientation field represented by points cloud. For each viewpoint, we project these directions onto the image plane and splat the points to obtain a 2D orientation map, using the head depth map to resolve hair–head occlusion. We trace strokes bidirectionally from uniform grid cell centers along local hair orientations, stopping at the background, the head region, or sharp directional changes that typically indicate hair partings. To promote even coverage, we maintain an occupancy map on the same grid, reject strokes that cross occupied cells, and mark cells traversed by accepted strokes as occupied. We also remove very short strokes to suppress noise.

To augment sketch diversity, we vary grid size to control stroke density, adjust stroke width and smoothness, and trace from a small subset of grid cells to generate partial sketches. We also randomly sample viewpoints with a bias toward frontal views. Together, these augmentations generate diverse sketches from limited ground-truth hairstyles while preserving their underlying geometry.

Orientation VAE Training We train the orientation VAE with orientation reconstruction loss and KL divergence term,

$$L_{\text{VAE}} = L_{\text{ori}} + \lambda_{\text{kl}} L_{\text{kl}}, \tag{6}$$

where $L_{\text{ori}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathbf{o}}_i - \mathbf{o}_i\|_2^2$ is the distance between predicted and ground-truth orientations at sample points inside the hair region. We set λ_{kl} to a small value to promote reconstruction accuracy.

Following HairLRM, we prioritize sampling rare hair orientations. We additionally use local orientation variance to sample regions with rapid directional changes more frequently. As training progresses, we gradually increase the sampling probabilities of both rare orientations and high-variance regions to improve reconstruction of complex hairstyles.

Diffusion Training We train the diffusion transformer in the orientation VAE latent space using flow matching. Given a latent code $\mathbf{z}_{\text{data}}$, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and time $t \sim \mathcal{U}(0, 1)$, we define the interpolated latent as $\mathbf{z}_t = (1-t)\boldsymbol{\epsilon} + t\mathbf{z}_{\text{data}}$. The network predicts the target velocity $\mathbf{z}_{\text{data}} - \boldsymbol{\epsilon}$, conditioned on $\mathbf{c}$:

$$L_{\text{FM}} = \mathbb{E}_{t,\mathbf{z}_{\text{data}},\boldsymbol{\epsilon},\mathbf{c}} \left[\|\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{c}) - (\mathbf{z}_{\text{data}} - \boldsymbol{\epsilon})\|_2^2 \right]. \tag{7}$$

We independently drop mesh and sketch conditions during training, using a null token for the dropped mesh condition. At inference, we apply decoupled classifier-free guidance:

$$\mathbf{v}_{\text{cfg}} = \mathbf{v}_\varnothing + s_m(\mathbf{v}_m - \mathbf{v}_\varnothing) + s_s(\mathbf{v}_s - \mathbf{v}_\varnothing), \tag{8}$$

where $\mathbf{v}_\varnothing$, $\mathbf{v}_m$, and $\mathbf{v}_s$ are the unconditional, mesh-only, and sketch-only predictions, respectively, and s_m and s_s control the mesh and sketch guidance strengths. Starting from Gaussian noise at $t = 0$, we integrate the guided velocity to $t = 1$ to generate a latent code.

To improve robustness, we add Gaussian noise ($\sigma = 0.1$) to the input mesh latents to reduce overfitting to the limited training data. We also perturb sketch camera directions with Gaussian jitter ($\sigma = 5^\circ$) to simulate camera estimation errors at inference time. At each training iteration, we sample a variable number of sketch views to support both single-view and multi-view inputs, and randomly select a precomputed augmented sketch for each view.

Inference For an in-the-wild image, we use an image generation model (Google, 2026) to synthesize corresponding multi-view images. We remove texture and illumination, decompose the resulting images into hair-only renderings, and reconstruct the hair meshes using an image-to-3D model (Tripo AI, 2026). We also extract hair-flow sketches from the input image. Alternatively, artists may draw directional strokes directly on a coarse 3D hair mesh as guidance; we project these strokes into the specified camera views to obtain posed sketches. We transform the mesh and sketches into spatially aligned conditions following the coordinate and camera conventions used during training. Given these conditions, we sample a strand latent, decode it into a volumetric orientation field, and recover explicit strands following a tracing procedure similar to CT2Hair (Shen et al., 2023). Further details are provided in the supplementary material.

4 Experiments

Implementation details Our StrandVAE contains 48M parameters and represents each hairstyle as a 2048×64 latent tensor. We train it for 200 epochs with a global batch size of 192 using AdamW with a learning rate of 5×10^{-4}. Our conditional DiT contains 16 transformer blocks with a hidden dimension of 512 and 8 attention heads. We encode the input hair mesh using the frozen encoder of a pretrained DORA-VAE (Chen et al., 2025). Its 2,048 mesh tokens are reduced to 1,024 tokens through learned attention pooling. For each 336×336 sketch, we extract 576 patch tokens using a frozen DINOv2 ViT-S/14 (Oquab et al., 2023) and concatenate features from layers 3, 6, and 9. We train the DiT for 300 epochs on four GPUs with a per-GPU batch size of 80 and a learning rate of 3.5×10^{-4}. During training, we randomly use 1, 2, or 4 sketch views. At inference, we use 30 Euler steps and extract explicit strands from the generated orientation field following a tracing procedure similar to CT2Hair (Shen et al., 2023). Further details are provided in the supplementary material.

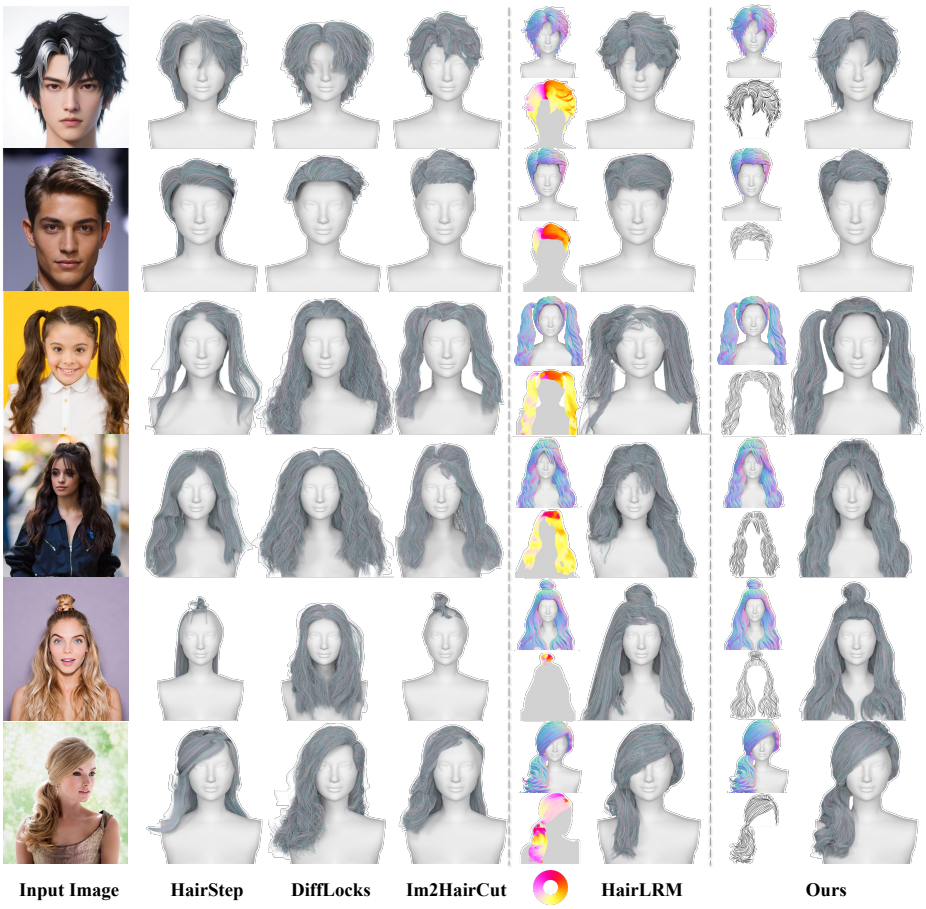

Figure 4: Qualitative comparison of single-view 3D hair reconstruction. HairLRM and HairFlow share the same input mesh, supplemented by HairStep orientation maps or sketches from image generation models, respectively. Inputs appear to the left of the corresponding strands. The color wheel below HairLRM illustrates the direction-to-color mapping.

Table 1: Quantitative comparison. Silhouette IoU (↑) is evaluated on 30 real images. Angular error (°, ↓) is measured between 3D orientation fields derived from reconstructed and ground-truth strands on 326 validation hairstyles.

Metric	HairStep	DiffLocks	Im2Haircut	HairLRM	**HairFlow (Ours)**
IoU ↑	0.625	0.583	0.773	0.855	0.912
Ori-Err ↓	56°	43°	35°	29°	22°

4.1 Comparisons

We compare HairFlow with HairStep Zheng et al. (2023), DiffLocks Rosu et al. (2025), Im2Haircut Sklyarova et al. (2025), and HairLRM Shen et al. (2026). HairStep lifts predicted strand and depth maps into 3D, while DiffLocks generates strand using image-conditioned diffusion over scalp-uv parameterization. Im2Haircut combines a transformer-based hair prior with Gaussian-splatting-based optimization. HairLRM optimizes the latent code of a pretrained hair autoencoder using a generated mesh and a HairStep strand map. Our method uses the same mesh and a sketch extracted by large image generation model as explicit geometric and directional constraints. Comparisons with HairGPT Luo et al. (2026) and Hao et al. (2026) are in the supplementary material.

Qualitative comparison Qualitative results can be found in in Fig 4. HairStep, Im2Haircut, and DiffLocks are trained on limited hair datasets and thus generalize poorly to complex real-world hairstyles, such as half-up buns and side ponytails. HairLRM involves meshes generated by a 3D foundation model, but still relies on 2D cues predicted by HairStep, which can be inaccurate for



Figure 5: **Effect of the sketch condition.** From left: input image, mesh, mesh-only result, sketch, and mesh+sketch result. Results within each row share the same mesh and random seed. Colors encode 3D tangent directions.

Table 2: Additional quantitative comparisons with HairLRM. We report mean angular errors (°, lower is better) for VAE reconstruction on the training and test sets, and for strand generation conditioned on meshes alone or with additional 2D cues.

Method	Training set	Validation set	Mesh Only	Mesh + 2D cues
HairLRM (Shen et al., 2026)	10.47	12.70	36.81°	29.52°
HairFlow (Ours)	**4.01**	**6.25**	27.43°	22.92°

complex hairstyles, as shown in the final two rows. Moreover, HairLRM fits a compact VAE latent to match the input mesh, which may not fully reproduce its geometry, leading to strand silhouettes visibly deviating from the input. In contrast, our method generalizes to diverse hairstyles while closely adhering to the geometry and strand flow specified by the input mesh and sketch.

Quantitative comparison We collect 30 real images and compute IoU between rendered strand silhouettes and input hair segmentation. For 3D evaluation, we hold out 326 validation hairstyles from training datasets, render them with a template head in Blender at angles from $-30°$ to $30°$, and use Nano Banana to complete the faces while keep the hair unchanged. The resulting images serve as inputs to baselines. At each spatial sample, we obtain directions from the nearest reconstructed strand point and the nearest ground-truth strand point, then average angular error. The quantitative results can be found in Table 1

Inference Time We benchmark available implementations and quote reported runtimes for HairGPT and Hao et al. (2026), whose public code we could not obtain. HairStep, DiffLocks, and Im2Haircut take 30.7, 48.8, and 378.3 s per image, respectively; HairGPT reports 30–60 s. HairLRM, Hao et al. (2026), and HairFlow additionally use foundation models to obtain geometric conditions. HairLRM takes $\sim$15 min: 3 min for mesh generation, 10 min for latent optimization, and 1–2 min for strand growing. Hao et al. (2026) reports $\sim$20 min, including multi-view image editing and $\sim$6 min for semantic processing and orientation propagation. HairFlow requires $\sim$3 min for mesh and sketch generation, 6.9 s for condition encoding, diffusion sampling, and field decoding, and 1–2 min for strand growing. Its core generation stage is approximately $87\times$ faster than HairLRM's reported latent optimization.

More quantitative comparison with HairLRM Tab. 2 provides additional quantitative comparisons with HairLRM (Shen et al., 2026). Our Orientation VAE reduces reconstruction angular error from $10.47°$ to $4.01°$ on the training set and from $12.70°$ to $6.25°$ on the validation set. This improvement benefits from a larger latent representation devoted entirely to the orientation field, without sharing capacity with SDF modeling. For mesh-conditioned generation on validation set, HairFlow achieves lower angular error both with and without 2D cues. Adding sketches further reduces our error from $27.43°$ to $22.92°$, compared with HairLRM's $29.52°$.

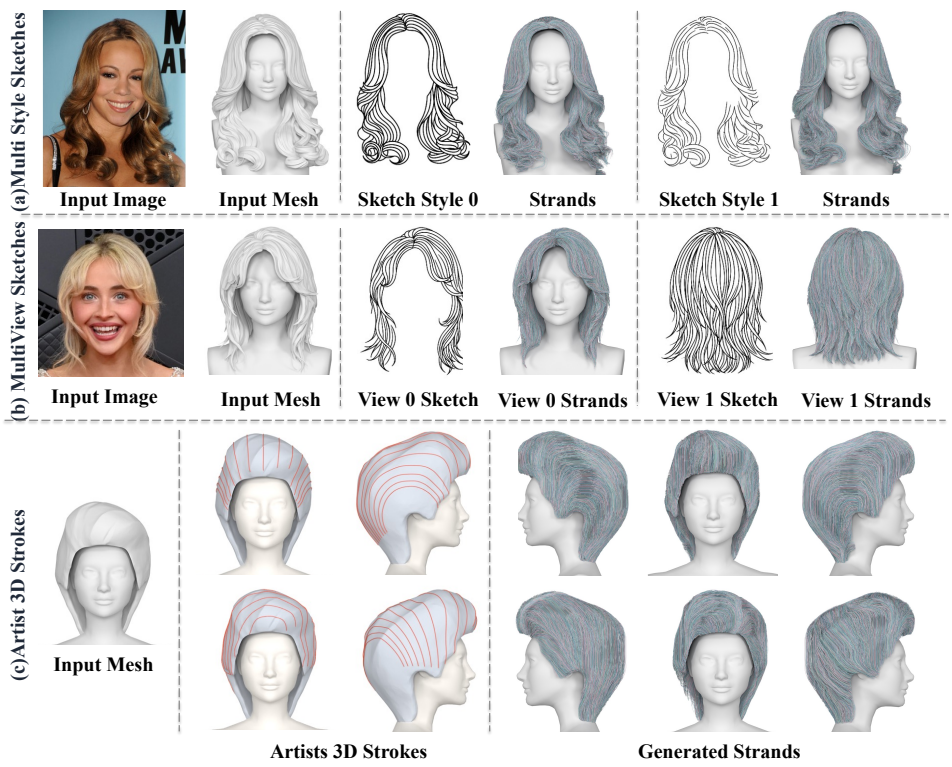

Figure 6: **Flexible Sketch Guidance:** (a) Sketches with different stroke widths and styles yield similar strand geometry. (b) Joint conditioning on sketches from two viewpoints produces a single hairstyle consistent with both views. (c) Two sets of artist-drawn 3D strokes on the same mesh are projected into multi-view sketches, guiding the generated strands to follow different flow patterns.

4.2 Ablations

Conditioning Ablation We compare mesh-only conditioning with joint mesh and sketch conditioning in Fig. 5. The mesh constrains the overall hair shape but does not explicitly specify strand directions. Adding sketches improves alignment with the hair flow in the input images, particularly in the second rows. Sketches also provide finer structural cues, producing more clearly separated hair clumps in the first row. The quantitative results in Table 2 further confirm that adding 2D sketches reduces angular error.

Flexible Sketch Guidance 6 illustrates the versatility of sketch guidance in HairFlow. In (a), two sketches of the same hairstyle differ substantially in appearance, yet produce similar strand structures, indicating robustness to stylistic variations while preserving local flow. In (b), front and back sketches jointly condition a single generation, yielding strands that follow the directional cues in both views. In (c), we project artist-drawn 3D strokes into multi-view sketches and use them as conditions. With the same input mesh, two sets of strokes produce distinct strand flow patterns aligned with the artist's intent, demonstrating directional control within a fixed global shape.

5 Conclusion

We presented HairFlow, a multimodal conditional diffusion framework for 3D hair strand modeling. It combines mesh–sketch cues from foundation models or artists with learned 3D strand priors to generate orientation fields without per-instance latent optimization. Meshes constrain global shape, while sketches guide local flow. An automatic sketch synthesis pipeline provides geometrically aligned training pairs. Experiments demonstrate improved input alignment, reduced inference time, and flexible control across modalities and styles.

REFERENCES

Menglei Chai, Lvdi Wang, Yanlin Weng, Yizhou Yu, Baining Guo, and Kun Zhou. Single-view hair modeling for portrait manipulation. *Acm Transactions on Graphics*, 31(4):1–8, 2012.

Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16251–16261. IEEE, 2025.

Google. Gemini 2.5 flash image (nano banana). Google AI for Developers documentation, 2026. Accessed: 2026-08-06.

Conghui Hao, Tao Huang, Yuefan Shen, Tongtong Wang, Zhongtian Zheng, and Kui Wu. Strand-based hairstyle generation via large reconstruction and multimodal models, 2026. URL https://arxiv.org/abs/2608.13679.

Chengan He, Xin Sun, Zhixin Shu, Fujun Luan, Sören Pirk, Jorge Alejandro Amador Herrera, Dominik L Michels, Tuanfeng Wang, Meng Zhang, Holly Rushmeier, et al. Perm: A parametric representation for multi-style 3d hair modeling. In *International Conference on Learning Representations*, volume 2025, pp. 84535–84561, 2025a.

Yuze He, Yanning Zhou, Wang Zhao, Jingwen Ye, Yushi Bai, Kaiwen Xiao, Yong-Jin Liu, Zhongqian Sun, and Wei Yang. Charm: Control-point-based 3d anime hairstyle auto-regressive modeling. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025b.

Jerry Hsu, Tongtong Wang, Zherong Pan, Xifeng Gao, Cem Yuksel, and Kui Wu. Real-time physically guided hair interpolation. *ACM Transactions on Graphics (TOG)*, 43(4):1–11, 2024.

Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Robust hair capture using simulated examples. *ACM Transactions on Graphics (TOG)*, 33(4):1–10, 2014.

Tao Huang, Yang Zhou, Daqi Lin, Junqiu Zhu, Lingqi Yan, and Kui Wu. Real-time level-of-detail strand-based rendering. In *Computer Graphics Forum*, volume 44, pp. e70181. Wiley Online Library, 2025.

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.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepmvshair: Deep hair modeling from sparse views. In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–8, 2022.

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.

Zixuan Lu, Tongtong Wang, Yuefan Shen, Zhongtian Zheng, Chenfanfu Jiang, Yin Yang, and Kui Wu. Haircs: Reconstructing strand-based hair from hair cards, 2026. URL https://arxiv.org/abs/2609.16465.

Haimin Luo, Min Ouyang, Lan Xu, and Jingyi Yu. Hairgpt: Strand-as-language autoregressive modeling for realistic 3d hairstyle synthesis. *ACM Transactions on Graphics (TOG)*, 45(4):1–14, 2026.

Stephen R Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley, and Pat Hanrahan. Light scattering from human hair fibers. *ACM Transactions on Graphics (TOG)*, 22(3):780–791, 2003.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Giljoo Nam, Chenglei Wu, Min H Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 155–164. IEEE, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

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 *European Conference on Computer Vision*, pp. 73–89. Springer, 2022.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10847–10857. IEEE, 2025.

Yuefan Shen, Changgeng Zhang, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepsketchhair: Deep sketch-based 3d hair modeling. *IEEE transactions on visualization and computer graphics*, 27 (7):3250–3263, 2020.

Yuefan Shen, Shunsuke Saito, Ziyan Wang, Olivier Maury, Chenglei Wu, Jessica Hodgins, Youyi Zheng, and Giljoo Nam. Ct2hair: High-fidelity 3d hair modeling using computed tomography. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023.

Yuefan Shen, Yican Dong, Xiufeng Huang, Zhongtian Zheng, Youyi Zheng, and Kui Wu. Hairlrm: Strand-based hair modeling via large reconstruction models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

Vanessa Sklyarova, Egor Zakharov, Otmar Hilliges, Michael J Black, and Justus Thies. Text-conditioned generative model of 3d strand-based human hairstyles. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4703–4712. IEEE, 2024.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10656–10665. IEEE, 2025.

Xiaokun Sun, Zeyu Cai, Ying Tai, Jian Yang, and Zhenyu Zhang. Strandhead: Text to hair-disentangled 3d head avatars using human-centric priors. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13393–13404. IEEE, 2025.

Kenji Tojo, Liwen Hu, Nobuyuki Umetani, and Hao Li. Strands2cards: Automatic generation of hair cards from strands. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Tripo AI. Introducing tripo h3.1: High detail model for production-ready assets. Official blog, June 2026. Accessed: 2026-08-06.

Cong Wang, Di Kang, Heyi Sun, Shenhan Qian, Zixuan Wang, Linchao Bao, and Song-Hai Zhang. Mega: Hybrid mesh-gaussian head avatar for high-fidelity rendering and head editing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26274–26284. IEEE, 2025a.

Junying Wang, Yuanlu Xu, Edith Tretschk, Ziyan Wang, Anastasia Ianina, Aljaz Bozic, Ulrich Neumann, and Tony Tung. Dgh: Dynamic gaussian hair. *Advances in Neural Information Processing Systems*, 38:45560–45594, 2026.

Yating Wang, Xuan Wang, Ran Yi, Yanbo Fan, Jichen Hu, Jingcheng Zhu, and Lizhuang Ma. 3d gaussian head avatars with expressive dynamic appearances by compact tensorial representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21117–21126. IEEE, 2025b.

Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. Hairclip: Design your hair by text and reference image. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18051–18060. IEEE, 2022.

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*, pp. 1526–1535, 2022.

Kui Wu and Cem Yuksel. Real-time hair mesh simulation. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 59–64, 2016.

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

Egor Zakharov, Vanessa Sklyarova, Michael Black, Giljoo Nam, Justus Thies, and Otmar Hilliges. Human hair reconstruction with strand-aligned 3d gaussians. In *European Conference on Computer Vision*, pp. 409–425. Springer, 2024.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

Na Zhang, Moran Li, Chengming Xu, Han Feng, Xiaobin Hu, Jiangning Zhang, Weijian Cao, Chengjie Wang, and Yanwei Fu. Stranddesigner: Towards practical strand generation with sketch guidance. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10296–10304, 2025.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12726–12735. IEEE, 2023.

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*, pp. 1–11, 2024.

Zhongtian Zheng, Tao Huang, Haozhe Su, Xueqi Ma, Yuefan Shen, Tongtong Wang, Yin Yang, Xifeng Gao, Zherong Pan, and Kui Wu. Auto hair card extraction for smooth hair with differentiable rendering. *ACM Transactions on Graphics (TOG)*, 44(6):1–13, 2025.

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):1–16, 2023.

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):1–15, 2024.

A More Experiments

More Comparisons We provide additional comparisons with HairGPT (Luo et al., 2026) and Hao et al. (2026). HairGPT autoregressively generates strand sequences, organizing hairstyles by semantic scalp regions and hierarchical strand structure. Hao et al. (2026) extract directional and semantic cues from multi-view renderings of an LRM-generated mesh, then propagate surface orientations into the hair volume using hand-crafted rules. Our generated strands follow the input hairstyles more faithfully than those of HairGPT and achieve visual quality comparable to Hao et al. (2026). Both our method and Hao et al. (2026) incur additional costs for 3D generation and image editing to obtain geometric conditions. Their pipeline reportedly takes approximately 20 min per example. In our pipeline, condition generation and strand growing together take approximately 4–5 min, while the core diffusion sampling and orientation-field decoding require only 6.9 s.



Figure 7: Qualitative comparisons with HairGPT (Luo et al., 2026) and Hao et al. (2026). Their input images and reconstruction results are taken from the respective papers, as code is unavailable.

Hair simulation To demonstrate the downstream usability of our generated strands, we import the reconstructed hairstyles into Unreal Engine and simulate their motion under wind. This experiment illustrates how our static reconstructions can be incorporated into a strand-based animation pipeline, extending their use beyond static rendering. The supplementary video shows the resulting hair motion.

Multi-view visualization We provide additional renderings of reconstructed strands from multiple viewpoints in Fig. 8, Fig. 9, and Fig. 10. These views reveal the overall hairstyle geometry and strand flow beyond the input viewpoint, allowing closer inspection of the reconstructed 3D structure.

B Additional Implementation Details

Orientation VAE. The orientation vae uses a hidden dimension of 384. Its encoder and latent transformer contain 8 and 16 layers, respectively, with 6 attention heads. For each training sample, we draw 32,768 orientation points and 32,768 salient points as encoder inputs, and supervise the decoder at 3,000 query points. The query points are sampled according to a combination of inverse orientation frequency and local directional variance. We use an orientation reconstruction loss together with a latent regularization term weighted by 10^{-4}. Training uses cosine learning-rate decay and gradient clipping with a maximum norm of 1.0.

Condition encoders. For mesh conditioning, we use the frozen encoder of the pretrained DORA-VAE-1.1 model, which produces 2,048 tokens of dimension 64. A learned attention-pooling module reduces them to 1,024 tokens. For sketch conditioning, we resize each view to 336×336 and use a frozen DINOv2 ViT-S/14. We concatenate its features from layers 3, 6, and 9. Each view contains $24 \times 24 = 576$ patch tokens. A 6D Plücker ray is computed for every patch and projected to a 32-dimensional camera embedding. Tokens from all views are jointly processed by a 6-layer transformer.

Training augmentation. The mesh latent is perturbed with Gaussian noise of standard deviation 0.1. We independently perturb the camera azimuth and elevation with Gaussian noise of standard deviation 5°. Mesh and sketch conditions are independently dropped with probability 0.1. At each iteration, the number of input sketch views is sampled from $\{1, 2, 4\}$.

Sketch synthesis. We render eight 512×512 orthographic orientation maps for each training hairstyle and trace them bidirectionally to produce sketches. We vary stroke spacing and width by sampling the occupancy-cell size from $\{8, 12, 16\}$ pixels and the stroke width from $\{1, 2, 3\}$ pixels. We also generate partial sketches by restricting stroke seeds to randomly sampled local regions.

Sampling and strand extraction. We scale Orientation VAE latents by 0.57 during diffusion training and undo this scaling after sampling. We use 30 Euler steps and decoupled classifier-free guidance:

$$\mathbf{v}_{\text{cfg}} = \mathbf{v}_{\varnothing} + s_m(\mathbf{v}_m - \mathbf{v}_{\varnothing}) + s_s(\mathbf{v}_{m,s} - \mathbf{v}_m), \tag{9}$$

where $\mathbf{v}_{\varnothing}$, $\mathbf{v}_m$, and $\mathbf{v}_{m,s}$ are the unconditional, mesh-conditioned, and jointly conditioned predictions, respectively. We use $s_m = 1.5$ and $s_s = 5.0$. The sampled latent is decoded on a regular grid inside the hair mesh. Explicit strands are then recovered from the volumetric orientation field using a tracing procedure similar to CT2Hair (Shen et al., 2023).

C Input Construction and Alignment

Input construction from a single image. Given an in-the-wild image, we first use an image generation model (Google, 2026) to synthesize front, back, left, and right views while preserving the identity and hairstyle of the reference. We then remove texture and illumination from the generated views to obtain clay-style renderings. These renderings are further decomposed into full-head and hair-only images. We feed the two sets of renderings to an image-to-3D model (Tripo AI, 2026) to reconstruct a full-head mesh and a coarse hair mesh, respectively. Hair-flow sketches are also extracted from the reference image or generated views while preserving their image-space layouts.

Mesh registration. Because the head and hair meshes are reconstructed independently, we first register the hair mesh to the full-head mesh using similarity ICP. We then align both meshes to the template bust in the coordinate system of the Orientation VAE. Specifically, we render the reconstructed head from the frontal and two nearby views, detect facial landmarks in the rendered images, and back-project them onto the mesh. A similarity transformation is estimated between these 3D landmarks and the corresponding landmarks on the template bust and is applied to both the head and hair meshes.

Camera and sketch alignment. We fit an orthographic camera by matching facial landmarks between the reconstructed head and the input view. For multi-view inputs, the remaining cameras are obtained by rotating the fitted frontal camera around the vertical axis. Image generation may introduce small view-dependent changes in scale and framing. We therefore estimate an additional 2D similarity transformation for each view by matching the projected hair-mesh silhouette with the corresponding hair-only rendering. The same transformation is applied to the hair-flow sketch. Finally, the aligned sketches are resized and arranged according to the camera convention used during training.

Artist-provided guidance. An artist can draw directional strokes directly on a coarse 3D hair mesh. When the mesh is transformed into the canonical Orientation VAE coordinate system, the same transformation is applied to the strokes. We then project the strokes through user-specified or automatically placed orthographic cameras and rasterize them as hair-flow sketches. Since the strokes are attached to the mesh surface and projected with the same cameras, the resulting mesh and sketch conditions are aligned by construction. The artist may draw strokes only in selected regions or provide guidance from a subset of views.

Image-generation prompts. We use task-specific prompts for multi-view generation, appearance removal, component decomposition, and sketch extraction. The multi-view prompt asks the image model to preserve the subject and hairstyle of the reference image, infer occluded hair regions, and follow a predefined four-view layout. A version of the prompt is:

> ***Multi-View Generation**: Reference Image 1 is the sole source of the character's identity, hairstyle, and visual style. Reference Image 2 is a four-panel template used only to control the viewpoints, head orientations, positioning, and layout. Generate four head views of the person from Reference Image 1: front, back, left profile, and right profile. Accurately infer and reconstruct the hair regions occluded in Reference Image 1. Preserve the facial identity, hairstyle, and visual style of Reference Image 1 in the final output. Use a pure white background. Remove everything below the neck and any objects or accessories on the head. Match the viewpoints, positioning, and four-panel layout of Reference Image 2 exactly. Maintain strict identity, hairstyle, and geometric consistency across all four views.*

Clay Conversion*: Generate a square four-panel grid showing a gray 3D clay sculpt. The sculpted details should be fine, smooth, and rich, with no rough surface texture. Use a pure white background. Match the geometry of the reference image exactly, while removing its original colors and photorealistic lighting and shadows. Keep the output dimensions identical to the reference image, and strictly align the position and viewpoint of each panel with the reference.*

Hair Isolation*: Isolate the hair in each panel of the four-panel reference image and remove all other elements. Keep the hair in exactly the same position, with the original viewpoints and four-panel layout unchanged.*

Sketch Extraction*: Input: A portrait image. Output: A hair geometry path map with the same dimensions as the input image, consisting only of pure white lines on an opaque black background. Remove the face, body, and all color, lighting, shading, and other appearance information. Each line must trace the complete path of a single hair strand. For each hair cluster, draw approximately three evenly spaced lines that clearly describe the cluster's 3D volume, width, and surface flow. Maintain visible gaps between lines within the same cluster. Ignore lighting and shadows entirely. Keep each line continuous, breaking it only where the strand is genuinely occluded. Every line must follow the hair's growth direction and align precisely with its position in the input image. Style: Vector graphic, flat SVG style, computer-generated splines, strictly uniform stroke width, monoline. Binary black-and-white image with hard edges and no anti-aliasing.*

D DISCUSSION

Limitations and future work While HairFlow handles a wide range of hairstyles, fine curls in Afro-textured hair and complex strand connectivity remain challenging, as volumetric orientation fields may smooth high-frequency details and do not explicitly represent connectivity. Reconstruction also depends on the input mesh: for example, a ponytail fused with the head can hinder the recovery of separated strands. Future work could explore spatially localized representations for interactive editing, enabling artists to modify selected hair regions through sketch guidance while preserving the rest of the hairstyle.

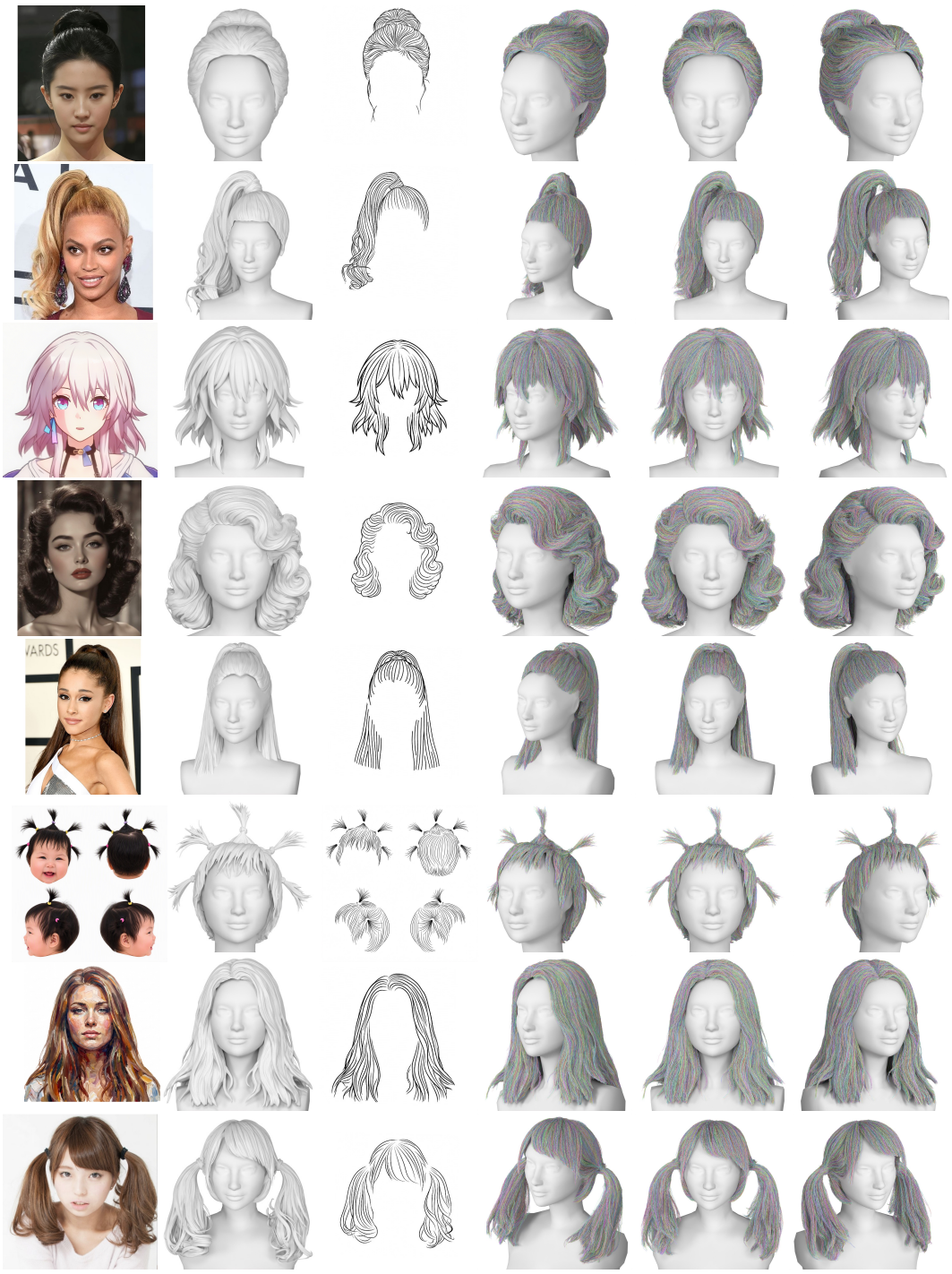

Figure 8: Multi-view renderings of reconstructed strands. Rows 3 and 7 use anime and oil-painting inputs, respectively; row 6 uses multi-view image inputs. The results demonstrate reconstruction across diverse hairstyles, image styles, and input views.



Figure 9: Additional multi-view renderings of reconstructed strands, showing hairstyle geometry and strand flow from different viewpoints.



Figure 10: Additional multi-view results. In the first row, an artist-provided text prompt is used to generate a hair mesh, and sketches are extracted from its rendered images, forming our unified geometric interface for strand generation.