Text-Conditioned Generative Model of 3D Strand-based Human Hairstyles

Vanessa Sklyarova[1,2]   Egor Zakharov[2]   Otmar Hilliges[2]   Michael J. Black[1]   Justus Thies[1,3]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]ETH Zürich, Switzerland   [3]Technical University of Darmstadt, Germany

Figure 1. Given a text description, our method produces realistic human hairstyles. The usage of a 3D strand-based geometry representation allows it to be easily incorporated into existing computer graphics pipelines for simulation and rendering [3, 7, 8].

Abstract

We present HAAR, a new strand-based generative model for 3D human hairstyles. Specifically, based on textual inputs, HAAR produces 3D hairstyles that could be used as production-level assets in modern computer graphics engines. Current AI-based generative models take advantage of powerful 2D priors to reconstruct 3D content in the form of point clouds, meshes, or volumetric functions. However, by using the 2D priors, they are intrinsically limited to only recovering the visual parts. Highly occluded hair structures can not be reconstructed with those methods, and they only model the "outer shell", which is not ready to be used in physics-based rendering or simulation pipelines. In contrast, we propose a first text-guided generative method that uses 3D hair strands as an underlying representation. Leveraging 2D visual question-answering (VQA) systems, we automatically annotate synthetic hair models that are generated from a small set of artist-created hairstyles. This allows us to train a latent diffusion model that operates in a common hairstyle UV space. In qualitative and quantitative studies, we demonstrate the capabilities of the proposed model and compare it to existing hairstyle generation approaches. For results, please refer to our project page[†].

1. Introduction

There has been rapid progress in creating realistic, animatable 3D face and head avatars from images, video, and text. What is still missing is hair. Existing methods typically represent hair with a coarse mesh geometry, implicit surfaces, or neural radiance fields. None of these representations are compatible with the strand-based models used by existing rendering systems and do not enable animation of the resulting avatars with natural hair dynamics. Modeling and generating realistic 3D hair remains a key bottleneck to creating realistic, personalized avatars. We address this

---

[†] https://haar.is.tue.mpg.de/

problem with HAAR (Hair: Automatic Animatable Reconstruction), which enables the generation of realistic and diverse hairstyles *based solely on text descriptions*. HAAR is the first text-driven generative model that produces a strand-based hair representation that can be immediately imported into rendering systems and animated realistically. This approach replaces the complex and time-consuming process of manually creating 3D hairstyles with a text interface that can be used by a novice to create high-quality hair assets.

Previous work exploits generative models as learned priors to create 3D strand-based hair from images, videos, or random noise. In particular, Neural Haircut [43] reconstructs high-fidelity hairstyles from smartphone video captures without any specialized equipment by leveraging a pre-trained generative diffusion model. However, their strand-based generative model does not provide control over the geometry of the resulting hairstyles, substantially limiting the range of applications. Recently, GroomGen [51] introduced an unconditional generative model of hair. In contrast, we propose the first text-conditioned generative model for strand-based hairstyles that can be used for automated and fast hair asset generation.

Text-conditioned generative models like Stable Diffusion [38] are widely used for image and video generation and can be used to generate 3D shape from text [5, 6, 12, 22, 25, 26, 28, 34, 35, 44, 45, 48] by exploiting Score Distillation Sampling (SDS) [34]. These methods convert textual descriptions into 3D assets that, when rendered into multiple views, align with generated 2D images via differentiable rendering. These methods represent 3D shapes either as meshes [5, 22, 35], point clouds [6, 44] or volumes [25, 26, 28, 34, 45]. In particular, TECA [48] demonstrates how hair can be generated from text using a neural radiance field [30], combined with a traditional mesh-based head model [20]. However, the inherent problem with these SDS-based solutions is that they only capture the outer visible surface of the 3D shape. Even volumetric representations do not have a meaningful internal hair structure [48]. Thus, they can not be used for downstream applications like animation in graphics engines [3, 7].

Instead, what we seek is a solution with the following properties: (1) the hair is represented using classical 3D strands so that the hairstyle is compatible with existing rendering tools, (2) hair is generated from easy-to-use text prompts, (3) the generated hair covers a wide range of diverse and realistic hairstyles, (4) the results are more realistic than current generative models based SDS. To this end, we develop a text-guided generation method that produces strand-based hairstyles via a latent diffusion model. Specifically, we devise a latent diffusion model following the unconditional model used in Neural Haircut [43]. A hairstyle is represented on the scalp of a 3D head model as a texture map where the values of the texture map correspond to the latent representation of 3D hair strands. The individual strands are defined in a latent space of a VAE that captures the geometric variation in the hair strand shape. To generate novel hair texture maps, we infer a diffusion network that takes a noise input and text conditioning. From the generated hair texture map, we can sample individual latent strands and reconstruct the corresponding 3D hair strands.

There are three remaining, interrelated, problems to address: (1) We need a dataset of 3D hairstyles to train the VAE and diffusion model. (2) We need training data of hairstyles with text descriptions to relate hairstyles to our representation. (3) We need a method to condition generated hair on text. We address each of these problems. First, we combine three different 3D hair datasets and augment the data to construct a training set of 9,825 3D hairstyles. Second, one of our key novelties is in how we obtain hairstyle descriptions. Here, we leverage a large vision-language model (VLM) [24] to generate hairstyle descriptions from images rendered from the 3D dataset. Unfortunately, existing visual question-answering (VQA) systems [19, 23, 24] are inaccurate and do not produce coherent hairstyle descriptions. To address these problems, we design a custom data-annotation pipeline that uses a pre-generated set of prompts that we feed into a VQA system [23] and produce final annotations by combining their responses in a single textual description. Finally, we train a diffusion model to produce the hair texture encoding conditioned on the encoding of textual hairstyle descriptions.

As Figure 1 illustrates, our strand-based representation can be used in classical computer graphics pipelines to realistically densify and render the hair [3, 7, 8]. We show how the latent representation of hair can be leveraged to perform various semantic manipulations, such as up-sampling the number of strands (resulting in better quality than the classical graphics methods) or editing hairstyles with text prompts. We perform quantitative comparisons with Neural Haircut as well as studies to understand which design choices are critical. In contrast to SDS-based methods like TECA, HAAR is significantly more efficient, requiring seconds instead of hours to generate a hairstyle.

Our contributions can be summarized as follows:
- We propose a first text-conditioned diffusion model for realistic 3D strand-based hairstyle generation,
- We showcase how the learned latent hairstyle representations can be used for semantic editing,
- We developed a method for accurate and automated annotation of synthetic hairstyle assets using VQA systems.

2. Related work

Recently, multiple text-to-3D approaches [5, 6, 12, 22, 25, 26, 28, 34, 35, 44, 45, 48] have emerged that were inspired by the success of text-guided image generation [36–

38, 40]. A body of work of particular interest to us is the one that uses image-space guidance to generate 3D shapes in a learning-by-synthesis paradigm. Initially, these methods used CLIP [37] embeddings shared between images and text to ensure that the results generated by the model adhere to the textual description [2, 10, 29]. However, the Score Distillation Sampling procedure (SDS) [34] has recently gained more popularity since it could leverage text-to-image generative diffusion models, such as Stable Diffusion [38], to guide the creation of 3D assets from text, achieving higher quality. Multiple concurrent methods employ this SDS approach to map textual description into a human avatar [4, 12, 17, 21, 48]. In particular, the TECA [48] system focuses on generating volumetric hairstyles in the form of neural radiance fields (NeRFs) [30]. However, these approaches can only generate the outer visible surface of the hair without internal structure, which prevents it from being used out-of-the-box in downstream applications, such as simulation and physics-based rendering. Moreover, the SDS-based reconstruction is notoriously slow and may require hours for a given text prompt. In contrast, HAAR is capable of generating and realistically rendering the hairstyles given textual prompts in less than a minute.

In contrast to the methods mentioned above, we also generate the hairstyles in the form of strands. Strand-accurate hair modeling has manifold applications in computer vision and graphics as it allows subsequent physics-based rendering and simulation using off-the-shelf tools [3, 7, 8]. One of the primary use cases for the strand-based generative modeling has historically been the 3D hair reconstruction systems [11, 18, 31, 39, 41–43, 46, 47, 49, 50]. Among the settings where it is most often used is the so-called one-shot case, where a hairstyle must be predicted using only a single image [11, 46, 49]. Approaches that tackle it leverage synthetic datasets of strand-based assets to train the models and then employ detailed cues extracted from the images, such as orientation maps [33], to guide the generation process. However, these systems are unsuitable for semantics-based or even unconditional generation of hairstyles, as they rely heavily on these cues for guidance. A group of methods that is more closely related to ours is Neural Haircut [43] and GroomGen [51], in which a synthetic dataset of hairstyle assets is leveraged to train an unconditional generative model [14, 16, 38]. While useful for regularizing multi-view hair reconstruction [43], the degree of control over the synthesized output in such methods is missing. Our work addresses the issue of controllability in generative models for hair and is the first one to provide strand-based hairstyle generation capabilities given textual descriptions.

3. Method

Given a textual description that contains information about hair curliness, length, and style, our method generates real-

istic strand-based hair assets. The resulting hairstyles can be immediately used in computer graphics tools that can render and animate the hair in a physically plausible fashion. Our pipeline is depicted in Figure 2. At its core is a latent diffusion model, which is conditioned on a hairstyle text embedding. It operates on a latent space that is constructed via a Variational Autoencoder (VAE) [16]. Following [39], this VAE is trained to embed the geometry of individual strands into a lower-dimensional latent space. During inference, the diffusion model generates this representation from Gaussian noise and the input text prompt, which is then upsampled to increase the number of strands and decoded using a VAE decoder to retrieve the 3D hair strands.

3.1. Hairstyle parametrization

We represent a 3D hairstyle as a set of 3D hair strands that are uniformly distributed over the scalp. Specifically, we define a hair map H with resolution 256×256 that corresponds to a scalp region of the 3D head model. Within this map, each pixel stores a single hair strand S as a polyline. As mentioned previously, our diffusion model is not directly operating on these 3D polylines, but on their compressed latent embeddings z. To produce z that encodes the strand S, we first convert the latter into the local basis defined by the Frenet frame of the face where the strand root is located. On this normalized data, we train a variational auto-encoder, which gives us access to an encoder $\mathcal{E}(S)$ and a decoder $\mathcal{G}(z)$. Using the encoder $\mathcal{E}(S)$, we encode the individual hair strands in the hair map H, resulting in a latent map Z that has the same spatial resolution. The decoded strand-based hair map is then denoted as $\hat{H}$. In summary, with a slight abuse of notation, the maps are related to each other as follows: $Z = \mathcal{E}(H)$, and $\hat{H} = \mathcal{G}(Z)$.

3.2. Conditional hair diffusion model

We use a pre-trained text encoder τ [19], that encodes the hairstyle description P into the embedding $\tau(P)$. This embedding is used as conditioning to the denoising network via a cross-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \quad (1)$$

where $Q = W_Q^{(i)} \cdot \phi_i(Z_t)$, $K = W_K^{(i)} \cdot \tau(P)$, $V = W_V^{(i)} \cdot \tau(P)$ with learnable projection matrices $W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}$. The denoising network is a 2D U-Net [13], where $\phi_i(Z_t)$ denotes i-th intermediate representations of the U-Net produced for the latent hair map Z_t at the denoising step t. For our training, we employ the EDM [14] formulation, following [43]. We denote the latent hair map with noise as $Z_t = Z + \epsilon \cdot \sigma_t$, where $\epsilon \sim \mathcal{N}(0, I)$, and σ_t is the noise strength. We then use a denoiser $\mathcal{D}$ to predict the output:

$$\mathcal{D}_\theta(Z_t, \sigma_t, P) = c_t^s \cdot Z_t + c_t^o \cdot \mathcal{F}_\theta\left(c_t^i \cdot Z_t, c_t^n, \tau(P)\right), \quad (2)$$
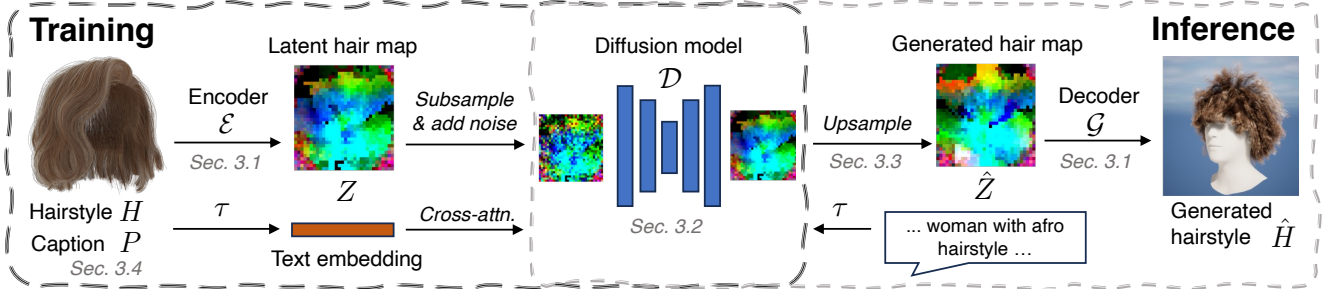
Figure 2. **Overview.** We present our new method for text-guided and strand-based hair generation. For each hairstyle H in the training set, we produce latent hair maps Z and annotate them with textual captions P using off-the-shelf VQA systems [23] and our custom annotation pipeline. Then, we train a conditional diffusion model $\mathcal{D}$ [14] to generate the *guiding strands* in this latent space and use a latent upsampling procedure to reconstruct dense hairstyles that contain up to a hundred thousand strands given textual descriptions. The generated hairstyles are then rendered using off-the-shelf computer graphics techniques [8].

where the c_t^s, c_t^o, c_t^i and c_t^n are the preconditioning factors for the noise level σ_t that follow [14], and $\mathcal{F}_\theta$ denotes a U-Net network. The optimization problem is defined as:

$$\min_\theta \ \mathbb{E}_{\sigma_t, \epsilon, Z, P} \left[\lambda_t \cdot \| \mathcal{D}_\theta(Z_t, \sigma_t, P) - Z \|_2^2 \right], \qquad (3)$$

where λ_t denotes a weighting factor for a given noise level.

3.3. Upsampling

Due to the limited amount of available 3D hairstyles, the diffusion model is trained on a downsampled latent hair map Z' with resolution 32×32 and, thus, only generates so-called "guiding hair strands". To increase the number of strands in the generated results, we upsample the latent hair map to the resolution of 512×512. A common way of upsampling a strand-based hairstyle to increase the number of strands is via interpolation between individual polylines.

In modern computer graphics engines [3, 7] multiple approaches, such as Nearest Neighbour (NN) and bilinear interpolation are used. Applying these interpolation schemes leads to over-smoothing or clumping results. In some more advanced pipelines, these schemes are combined with distance measures based on the proximity of strand origins or the similarity of the curves. Additionally, Blender and Maya [3, 7] introduce an option of adding noise into the interpolation results to further prevent clumping of the hair strands and increase realism. However, the described interpolation procedure requires a lot of manual effort and needs to be done for each hairstyle separately to obtain optimal parameters and resolve undesired penetrations.

In this work, we propose an automatic approach with interpolation of the hairstyle in *latent space* by blending between nearest neighbor and bilinear interpolation schemes. In this way, we aim to preserve the local structure of strands near a partition and apply smoothing in regions with similar strand directions. To calculate the blending weights, we first compute the cosine similarity between neighboring 3D hair

strands on the mesh grid and apply the non-linear function $f(\cdot)$ to control the influence of the particular interpolation type, which we empirically derived to be as follows:

$$f(x) = \begin{cases} 1 - 1.63 \cdot x^5 & \text{where } x \le 0.9 \\ 0.4 - 0.4 \cdot x & x > 0.9, \end{cases} \qquad (4)$$

where x is the cosine similarity. Our final interpolation for each point on the mesh grid is defined as a blending between the nearest neighbor and bilinear interpolations with the weight $f(x)$ and $(1 - f(x))$ correspondingly. Our upsampling method ensures that in the vicinity of a partition, the weight of the nearest neighbor decreases linearly, and then diminishes at a polynomial rate. As a result, we obtain realistic geometry in the regions with low similarity among strands. On top of that, we add Gaussian noise to the interpolated latents to increase the hair strands diversity, resulting in a more natural look.

3.4. Data generation

3D hairstyle data. For training and evaluating the diffusion model, we use a small artist-created hairstyle dataset, that consists of 40 high-quality hairstyles with around 100,000 strands. To increase the diversity, we combine it with two publicly available datasets: CT2Hair [42] and USC-HairSalon [11] that consist of 10 and 343 hairstyles, respectively. We align the three datasets to the same parametric head model and additionally augment each hairstyle using realistic squeezing, stretching, cutting, and curliness augmentations. In total, we train the model on 9,825 hairstyles.

Hairstyle description. As these 3D hairstyles do not come with textual annotations, we use the VQA model LLaVA [23, 24] to automatically produce hairstyle descriptions from a set of predefined questions (see Figure 3). To do that, we first render all collected hairstyles using Blender [7] from frontal and back camera views. We use
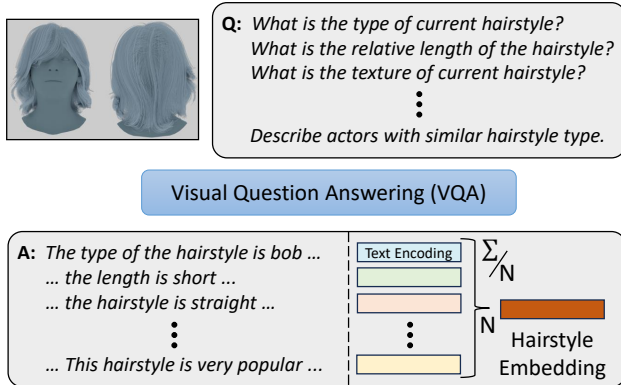
Figure 3. **Dataset collection.** Rendered from frontal and back view hairstyles along with a predefined set of questions Q are sent through VQA [23, 24] to obtain hairstyle description, which is further encoded using frozen text encoder network [19].

the standard head model and neutral shading for hairstyles to prevent bias to any particular type of hairstyle because of color or gender information. With the help of Chat-GPT [32], we design a set of prompts, that include specific questions about length, texture, hairstyle type, bang, etc., as well as a set of general questions about historical meaning, professional look, occasions for such hairstyle, celebrities with similar type to increase generalization and variability of our conditioning model. We then use a random subset of these prompts for each hairstyle in the dataset to increase the diversity of annotations. For a full list of prompts that were used, please refer to the suppl. material.

The quality of visual systems is highly restricted by the diversity of data used during training. We have observed in our experiments that the accuracy of the produced hair captions is relatively low, or they contain very broad descriptions. In particular, we have noticed that the existing VQA systems have problems accurately reasoning about the hair length or the side of the parting. To improve the quality of VQA answers, similarly to [52], we add an additional system prompt *"If you are not sure say it honestly. Do not imagine any contents that are not in the image"*, which decreases the likelihood of the model hallucinating its responses. Further, we have observed that the VQA system works better when it does not use information from the previous answers. That allows us to not accumulate erroneous descriptions during the annotation session. We have also observed that the LLAVA model is biased toward confirming the provided descriptions instead of reasoning, so introducing a set of choices to the prompts substantially improves the results. Finally, we calculate the embeddings of the resulting hairstyle descriptions P using a BLIP encoder τ for both frontal and back views and average them to produce the conditioning used during training.

3.5. Training details

For training, we sample a batch of hairstyles at each iteration, align them on a mesh grid of 256×256 resolution, and, then, subsample it to a size of 32×32. By training the diffusion model on these subsampled hairstyles we improve convergence and avoid overfitting. To accelerate the training, we use the soft Min-SNR [9] weighting strategy. It tackles the conflicting directions in optimization by using an adaptive loss weighting strategy. To evaluate the performance, we utilize an Exponential Moving Average (EMA) model and Euler Ancestral Sampling with 50 steps. The whole method is trained for about 3 days on a single NVIDIA A100, which corresponds to 160,000 iterations. Additional details are in the suppl. material.

4. Experiments

4.1. Evaluation

We compare our method against competing approaches for generative hair modeling: TECA [48] and Neural Haircut [43]. TECA creates a compositional avatar that includes separate geometries for hair, body, and cloth using only a text description. This method represents hair using neural radiance fields (NeRF) [30] and focuses on the visual quality of generated avatars, not geometry reconstruction. Moreover, it takes multiple hours to generate a single sample using TECA because they rely on Score Distillation Sampling [34]. In our case, we concentrate on physically plausible geometry for the hair and require around 4.3 seconds to generate a hairstyle. Neural Haircut focuses on the reconstruction of realistic 3D hairstyles with a strand-based representation using monocular video or multi-view images captured under unconstrained lighting conditions. In this work, authors exploit a diffusion model to obtain prior knowledge for better reconstruction quality. In contrast to our approach, the quality of their diffusion model is limited by the amount of data, the size of the model architecture, and the chosen training strategy. Furthermore, it is unconditional and cannot control the generated hairstyles.

Quality of unconditional diffusion. To compare the quality of the unconditional diffusion model, we re-train Neural Haircut [43] on the same training data and with the same scalp parametrization as our method. Also, we provide results of Neural Haircut with our novel training scheme (denoted as NH++), that includes signal-to-noise ratio weighting (SNR) and cosine noise sampling. Finally, we show the importance of this components in our model by presenting results without SNR, and without cosine noise scheduling.

We evaluate the distance of the generated hairstyles to the training distribution using Minimum Matching Distance (MMD) [1] as well as coverage (Cov) [1] metrics. We use the 1-Nearest Neighbor accuracy (1-NNA) [27] metric, which is a leave-one-out accuracy of the 1-NN classifier
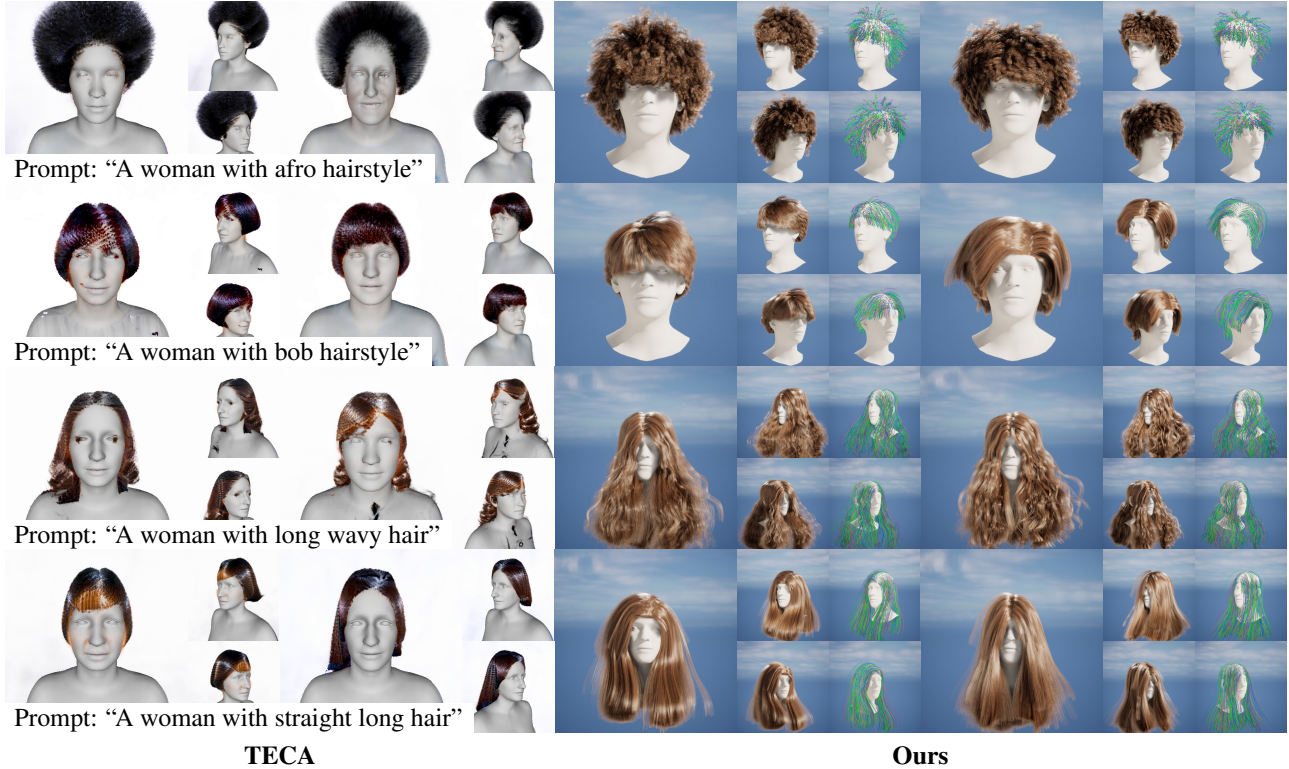
Figure 4. **Comparison.** Qualitative comparison of conditional generative models. We show several generations of TECA [48] and our model. For our results, we visualize the geometry obtained before (shown in pseudocolor) and after upsampling. Our model generates more diverse samples with higher-quality hairstyles. It is also worth noting that TECA, in some cases, does not follow the input descriptions well, producing short hair instead of long hair (bottom row). Digital zoom-in is recommended.

that assesses if two provided distributions are identical. The best quality is achieved for values closer to 0.5. Given two datasets of generated and reference hairstyles denoted as S_g and S_r, where $|S_g| = |S_r|$, we define the metrics as:

$$\text{MMD}(S_g, S_r) = \frac{1}{|S_r|} \sum_{y \in S_r} \min_{x \in S_g} D(x, y) \quad (5)$$

$$\text{COV}(S_g, S_r) = \frac{1}{|S_r|} |\{\arg \min_{y \in S_r} D(x, y) | x \in S_g\}| \quad (6)$$

$$1 - \text{NNA}(S_g, S_r) = \frac{\sum_{x \in S_g} \mathbb{I}[N_x \in S_g] + \sum_{y \in S_r} \mathbb{I}[N_y \in S_r]}{|S_g| + |S_r|}, \quad (7)$$

where $\mathbb{I}(\cdot)$ is an indicator function, N_F is the nearest neighbor in set $S_r \cup S_g \setminus F$ and D is the squared distance between distributions, computed in the latent space of the VAE. Lastly, we calculate FID using renders of the six views for 1,670 generated hairstyles. Based on these metrics, we show a comparison in Table 1. HAAR generates samples closer to the ground-truth distribution with higher diversity.

We conducted a user study, where participants were presented 40 randomly sampled hairstyle pairs from Neural Haircut [43] and HAAR, answering the question *"Which*

Method	MMD↓	COV↑	1-NNA → 0.5	FID↓
Neural Haircut [43]	31507.7	0.18	0.34	75.5
NH++	29329.4	0.14	0.48	42.6
HAAR wo cosine	27410.6	0.08	0.55	62.1
HAAR wo SNR	21499.2	0.19	0.56	44.7
HAAR	21104.9	0.2	0.55	36.9

Table 1. **Comparison of unconditional diffusion models.** Our method generates samples with better quality and diversity.

hairstyle from the presented pair is better?". From more than 1,200 responses, HAAR was preferred in 87.5%.

Quality of conditional diffusion. We compare the quality of our conditional generation with TECA [48], using various prompts with several random seeds to obtain the hairstyle that follows the desired text input. The qualitative comparison can be seen in Figure 4. While TECA produces great conditioning results most of the time, some severe artifacts are noticeable in the hair region. Furthermore, the diversity of generations is limited, and we see some failure cases even for simple prompts like *"A woman with straight long hair"*. With HAAR, we provide a way to obtain detailed physically plausible geometry with large variations.
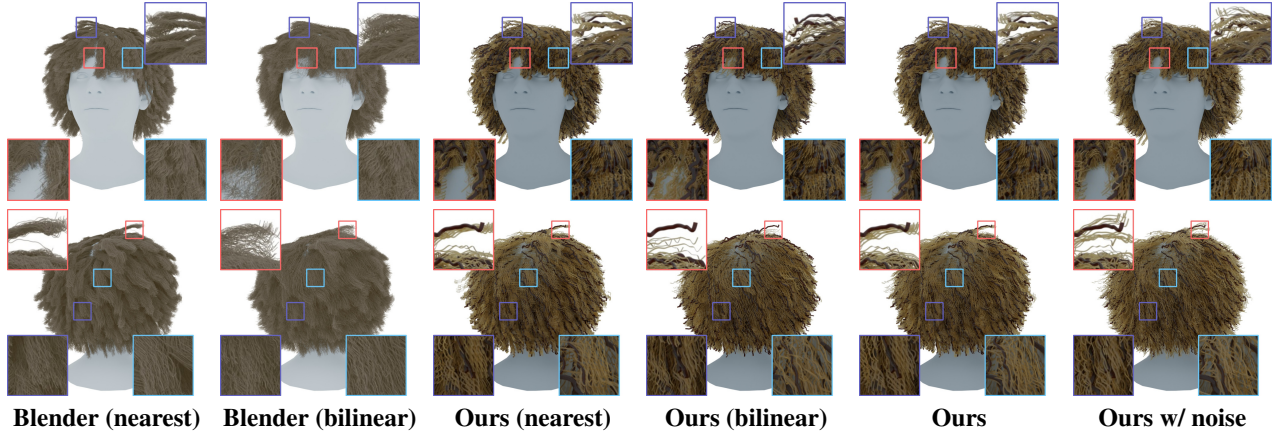
Figure 5. **Upsampling.** Comparison of different upsampling schemes used to interpolate between guiding strands (shown in dark color). For visualization purposes here we show around 15,000 strands. Blender interpolation is obtained in 3D space, while Ours is computed in latent space. Using the Nearest Neighbour in both variants produces better accuracy according to the guiding strand geometry (shown in dark color), but it results in an unrealistic global appearance. The bilinear schemes lead to the penetration of averaged hair strands and the loss of structure of the original guiding strands. Blending both these methods resolves proposed issues and results in realistic renders. Adding additional noise in latent space further increases realism and helps to get rid of the grid structure.

Experiment	frontal only	one question	HAAR	HAAR-10%	HAAR-200%
CSIM↑	0.186	0.18	0.189	0.187	0.191

Table 2. Ablation on conditioning types and number of samples.

4.2. Ablation study

Data collection method. To improve the accuracy of the question-answering model and reduce the number of incorrect annotations we employed a few tricks. We aggregate the answers over multiple query questions and filter the ones for which the model shows high uncertainty. We also average descriptions for frontal and back views. In Table 2, we evaluate the performance of our resulting model with either of these features removed. We provide additional results for HAAR trained on 10% (HAAR-10%) and on 200% (HAAR-200%) of the training samples.

We evaluate the performance using semantic matching between text and generated 3D hairstyles. Specifically, we use CLIP [37] and compute the cosine distance between images and their respective text prompts. To do that, we generate 100 hairstyles for 10 different prompts and then render from a frontal view using Blender [7].

Upsampling scheme. We ablate the performance of different upsampling schemes needed to obtain a full hairstyle from a set of guiding strands, which can be seen in Figure 5. There is no one-to-one correspondence and during interpolation, a lot of artifacts can occur. The most common artifact is a visible *grid structure* which appears when using a Nearest Neighbour (NN) strategy. Bilinear interpolation leads to *scalp penetrations* due to averaging the nearest strands on top of the head, and it deteriorates the local shape of curls.

The computer graphics engines, such as Blender [7] and Maya [3], either do not provide enough control or require a lot of manual effort in setting up the optimal parameters for each hairstyle separately. We find that the combination of NN and Bilinear using our proposed scheme leads to the best-looking results of renders. Furthermore, adding noise in the latent space results in more realistic hairstyles. Note, for visualization we show an example with a reduced density of around 15,000 strands; increasing it leads to less bald regions, especially, in the region of a partition.

4.3. Hairstyle editing

Similar to Imagic [15], we do text-based hairstyle editing, see Figure 6. Given an input hairstyle and a target text that corresponds to the desired prompt, we edit the hairstyle in a way that it corresponds to the prompt while preserving the details of the input hairstyle. To do that we first do textual inversion of the input hairstyle. We obtain e_{tgt} that corresponds to the target prompt P. After optimizing it with a fixed diffusion model $\mathcal{D}_\theta$ using a reconstruction loss, we acquire e_{opt}. Conditioning on the obtained text embedding e_{opt} does not lead to the same target hairstyle. So, to provide a smooth transition, we freeze e_{opt} and fine-tune $\mathcal{D}_\theta$. Finally, we linearly interpolate between e_{tgt} and e_{opt}. For more information, please refer to the supplemental material.

4.4. Limitations

The quality of generated hairstyles is limited by the variety and quality of our dataset, in terms of both the diversity of geometry assets and the accuracy of textual annotations. The main failure cases include the generation of hairstyles
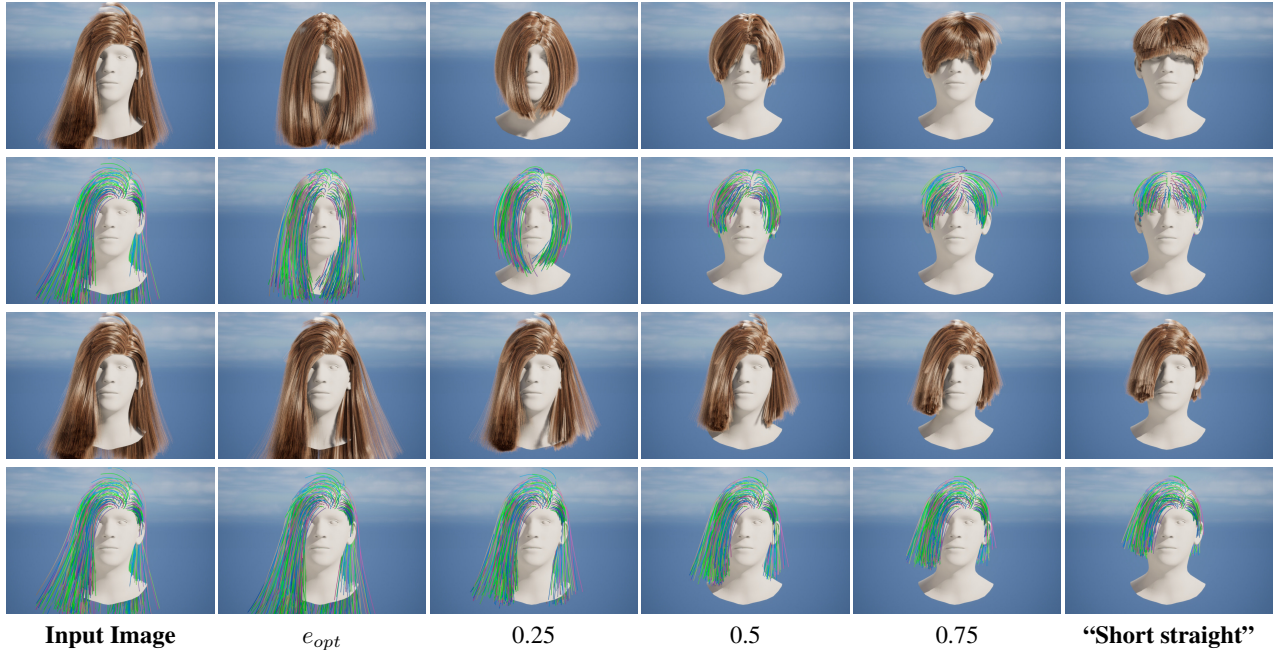
Figure 6. **Hairstyle editing.** Similar to Imagic [15], we edit the input using a text prompt without additionally fine-tuning the diffusion model (first two rows) and with (second two rows). Fine-tuning results in smoother editing and better preservation of input hairstyle.

Figure 7. **Limitations.** Our failure cases include penetration into the scalp region (left), which in principle can be resolved in a post-processing step. Additionally, for the afro hairstyles (right), the degree of strands' curliness needs to be increased.

with scalp interpenetrations and lack of curliness for some extreme hairstyles, see Figure 7. In theory, these limitations can be addressed with a dataset that contains more diverse samples of curly hairstyles, as well as human-made annotations. Especially, when used in a physics simulation, the interpenetrations can be resolved in a postprocessing step. Another limitation of our method is that we only consider geometry, we do not generate the hair color and texture which would be an interesting direction for future work.

5. Conclusion

We have presented HAAR, the first method that is able to conditionally generate realistic strand-based hairstyles using textual hairstyle descriptions as input. Not only can such

a system accelerate hairstyle creation in computer graphics engines, but it also bridges the gap between computer graphics and computer vision. For computer graphics, generated hairstyles could be easily incorporated into tools like Blender for hair editing and physics-based animation. For computer vision, our system can be used as a strong prior for the generation of avatars or to create synthetic training data of realistic hairstyles. While being limited by data, we think that this method is a first step in the direction of controllable and automatic hairstyle generation.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *Proceedings of the 35th International Conference on Machine Learning*, pages 40–49. PMLR, 2018. 5

[2] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. ClipFace: Text-guided Editing of Textured 3D Morphable Models. In *ArXiv preprint arXiv:2212.01406*, 2022. 3

[3] Autodesk, INC. Maya. 1, 2, 3, 4, 7

[4] Yukang Cao, Yan-Pei Cao, Kai Han, Ying Shan, and Kwan-Yee K Wong. Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. *arXiv preprint arXiv:2304.00916*, 2023. 3

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[6] Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2023. 2

[7] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2023. 1, 2, 3, 4, 7

[8] Epic Games. Unreal engine. 1, 2, 3, 4

[9] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7441–7451, 2023. 5

[10] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 3

[11] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (TOG)*, 34:1 – 9, 2015. 3, 4

[12] Yangyi Huang, Hongwei Yi, Yuliang Xiu, Tingting Liao, Jiaxiang Tang, Deng Cai, and Justus Thies. TeCH: Text-guided Reconstruction of Lifelike Clothed Humans. In *International Conference on 3D Vision (3DV)*, 2024. 2, 3

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 3

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3, 4

[15] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 7, 8

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 3

[17] Nikos Kolotouros, Thiemo Alldieck, Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Fieraru, and Cristian Sminchisescu. Dreamhuman: Animatable 3d avatars from text. 2023. 3

[18] Zhiyi Kuang, Yiyang Chen, Hongbo Fu, Kun Zhou, and Youyi Zheng. Deepmvshair: Deep hair modeling from sparse views. *SIGGRAPH Asia 2022 Conference Papers*, 2022. 3

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2, 3, 5

[20] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 2

[21] Tingting Liao, Hongwei Yi, Yuliang Xiu, Jiaxiang Tang, Yangyi Huang, Justus Thies, and Michael J Black. Tada! text to animatable digital avatars. *ArXiv*, 2023. 3

[22] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 2, 4, 5

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2, 4, 5

[25] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion, 2023. 2

[26] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2

[27] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017. 5

[28] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360 reconstruction of any object from a single image. In *CVPR*, 2023. 2

[29] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021. 3

[30] 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. In *ECCV*, 2020. 2, 3, 5

[31] Giljoo Nam, Chenglei Wu, Min H. Kim, and Yaser Sheikh. Strand-accurate multi-view hair capture. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 155–164, 2019. 3

[32] *ChatGPT*. OpenAI, 2023. 5

[33] Sylvain Paris, Héctor M. Briceño, and François X. Sillion. Capture of hair geometry from multiple images. *ACM SIGGRAPH 2004 Papers*, 2004. 3

[34] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2, 3, 5

[35] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023. 2

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 3, 7

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[39] 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. *European Conference on Computer Vision (ECCV)*, 2022. 3

[40] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. 2022. 3

[41] Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 3d hair synthesis using volumetric variational autoencoders. *ACM Transactions on Graphics (TOG)*, 37:1 – 12, 2018. 3

[42] 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*, 42(4): 1–13, 2023. 4

[43] 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. 2, 3, 5, 6

[44] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[45] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023. 2

[46] 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. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1516–1525, 2022. 3

[47] Lingchen Yang, Zefeng Shi, Youyi Zheng, and Kun Zhou. Dynamic hair modeling from monocular videos using deep neural networks. *ACM Transactions on Graphics (TOG)*, 38: 1 – 12, 2019. 3

[48] H. Zhang, Y. Feng, P. Kulits, Y. Wen, J. Thies, and M. J. Black. Teca: Text-guided generation and editing of compositional 3d avatars. *arXiv*, 2023. 2, 3, 5, 6

[49] 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*, pages 12726–12735, 2023. 3

[50] 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 *European Conference on Computer Vision*, 2018. 3

[51] Yuxiao Zhou, Menglei Chai, Alessandro Pepe, Markus Gross, and Thabo Beeler. GroomGen: A high-quality generative hair model using hierarchical latent representations. 2023. 2, 3

[52] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *ArXiv*, abs/2303.06594, 2023. 5