Supplementary Materials for 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

1. Implementation and training details

Hairstyle diffusion model. For conditional diffusion model, we use the U-Net architecture from [13] with the following parameters: $image\_size = 32 \times 32$, $input\_channels = 64$, $num\_res\_blocks = 2$, $num\_heads = 8$, $attention\_resolutions = (4, 2, 1)$, $channel\_mult = (1, 2, 4, 4)$, $model\_channels = 320$, $use\_spatial\_transformer = True$, $context\_dim = 768$, $legacy = False$.

Our training pipeline uses the EDM [5] library and we optimize the loss function using AdamW [10] with $learning\_rate = 10^{-4}$, $\beta = [0.95, 0.999]$, $\epsilon = 10^{-6}$, $batch\_size = 8$, and $weight\_decay = 10^{-3}$.

List of prompts. Below we include the list of prompts used during data annotation using a VQA model. After each of the prompts, we add *'If you are not sure say it honestly. Do not imagine any contents that are not in the image. After the answer please clear your history.'* to the input.

- 'Describe in detail the bang/fringe of depicted hairstyle including its directionality, texture, and coverage of face?'
- 'What is the overall hairstyle depicted in the image?'
- 'Does the depicted hairstyle longer than the shoulders or shorter than the shoulders?'
- 'Does the depicted hairstyle have a short bang or long bang or no bang from frontal view?'
- 'Does the hairstyle have a straight bang or Baby Bangs or Arched Bangs or Asymmetrical Bangs or Pin-Up Bangs or Choppy Bangs or curtain bang or side swept bang or no bang?'
- 'Are there any afro features in the hairstyle or no afro features?'
- 'Is the length of the hairstyle shorter than the middle of the neck or longer than the middle of the neck?'
- 'What are the main geometry features of the depicted hairstyle?'
- 'What is the overall shape of the depicted hairstyle?'
- 'Is the hair short, medium, or long in terms of length?'
- 'What is the type of depicted hairstyle?'
- 'What is the length of hairstyle relative to the human body?'
- 'Describe the texture and pattern of hair in the image.'
- 'What is the texture of depicted hairstyle?'
- 'Does the depicted hairstyle is straight or wavy or curly or kinky?'
- 'Can you describe the overall flow and directionality of strands?'
- 'Could you describe the bang of depicted hairstyle including its directionality and texture?'

Figure 1. **Dataset.** Hairstyles used during training (row 1) and upsampled versions (row 2).

- 'Describe the main geometric features of the hairstyle depicted in the image.'
- 'Is the length of a hairstyle buzz cut, pixie, ear length, chin length, neck length, shoulder length, armpit length or mid-back length?'
- 'Describe actors with similar hairstyle type.'
- 'Does the hairstyle cover any parts of the face? Write which exact parts.'
- 'In what ways is this hairstyle a blend or combination of other popular hairstyles?'
- 'Could you provide the closest types of hairstyles from which this one could be blended?'
- 'How adaptable is this hairstyle for various occasions (casual, formal, athletic)?'
- 'How is this hairstyle perceived in different social or professional settings?'
- 'Are there historical figures who were iconic for wearing this hairstyle?'
- 'Could you describe the partition of this hairstyle if it is visible?'

Text-based models. For the VQA, we found that "LLaVA-v1.5" model [8, 9] produces the best text descriptions with a relatively low hallucination rate. In Table 1, we experimented with different text encoder models. We used "ViT-L/14" configuration for CLIP [12] and "blip_feature_extractor" from [6] library for BLIP [7]. In the ablation experiment, we compare its result with an optimizable Transformer [14] build on top of the pre-trained BERT-Tokenizer [2] from the transformers [15] library with configuration "bert-base-uncased". For the Transformer network, we use BERTEmbedder from [13] with $n\_layer = 6$, $max\_seq\_len = 256$, $n\_embed = 640$.

Dataset. We provide several examples of the dataset

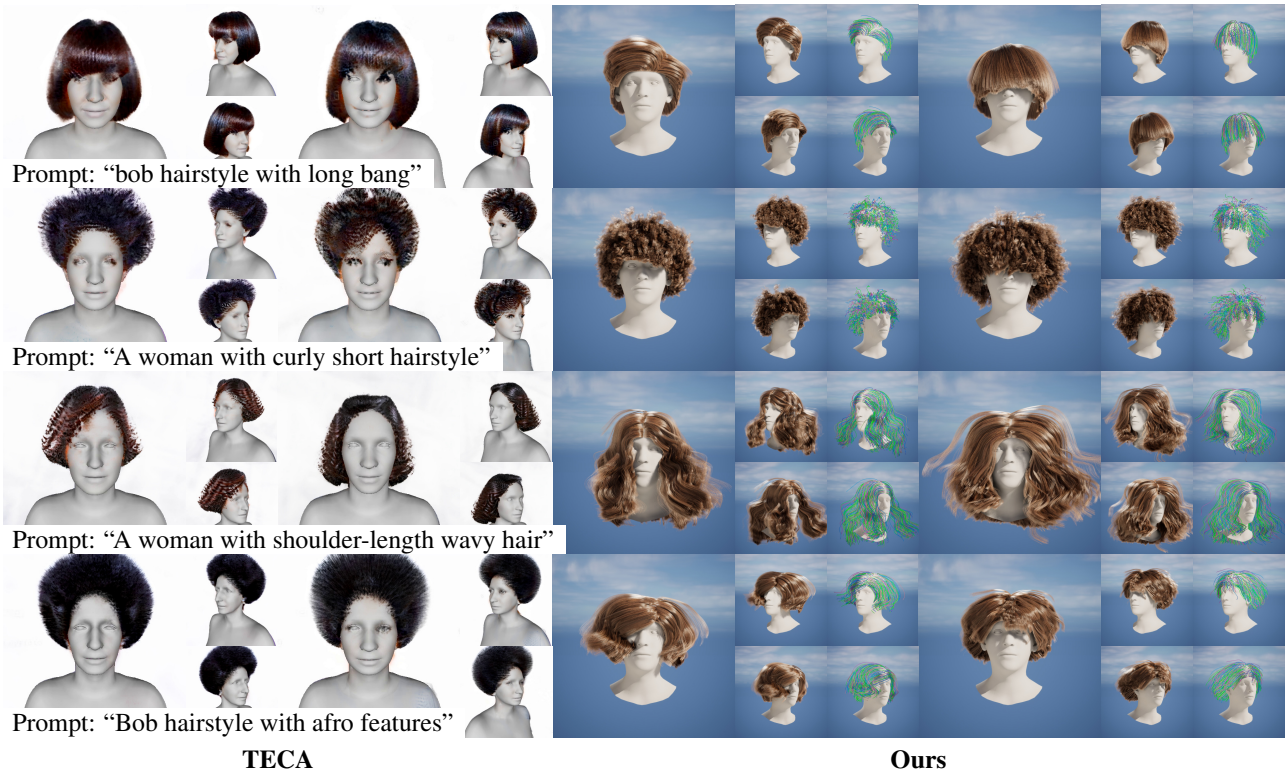

Figure 2. **Comparison.** Extended comparison with TECA. Our method produces higher quality samples with greater diversity than ones generated in TECA, and our representation allows the animation of the hair in a physics simulator.

samples used during training and the upsampled versions, see Figure 1.

2. Additional Ablations and Results

Qualitative comparison. We show an extended comparison with TECA [16] with more complex prompts that show the compositional abilities of the models (see Figure 2).

Importance of classifier-free-guidance. To improve the sample quality of the conditional model, we use classifier-free-guidance [4]. During training, we optimize conditional and unconditional models at the same time, by using text embedding with zeros in 10% of cases. During inference, we fix the random seed and show changes in sample quality, sweeping over the guidance strength w. As we can see in Figure 4, higher weights improve the strength of conditional prompts, but increasing it too much leads to out-of-distribution samples with a high degree of inter-head penetrations. In our experiments, we fix the guidance weight to $w = 1.5$.

Architecture of text encoder. The quality of the conditional diffusion model for hairstyle generation is highly dependent on the quality of the text encoder network $\tau(\cdot)$. We ablate the performance of the conditional generation using pre-trained and frozen encoders, such as CLIP [12], BLIP [7] as well as a trained transformer network [14] implemented on top of a pre-trained BertTokenizer [2].

Text encoder	CLIP	BLIP	Transf.	Reference
CSIM	0.174	0.189	0.172	0.206

Table 1. **Conditioning.** Ablation on different conditioning schemes. With BLIP text encoder, we obtain better conditioning compared to CLIP and trainable Transformer network.

Figure 3. **Upsampling.** Results of upsampling in latent space (left image) compared to upsampling in 3D space (right image) with our final interpolation scheme. Digital zoom-in is recommended.

The intuition behind training additional networks for text encoding is that the quality of pre-trained encoders may be limited for a particular task (for example some specific hairstyle types), which results in wrong correlations between words and deteriorates the quality of the diffusion model.

Table 1 shows that the BLIP text encoder provides the

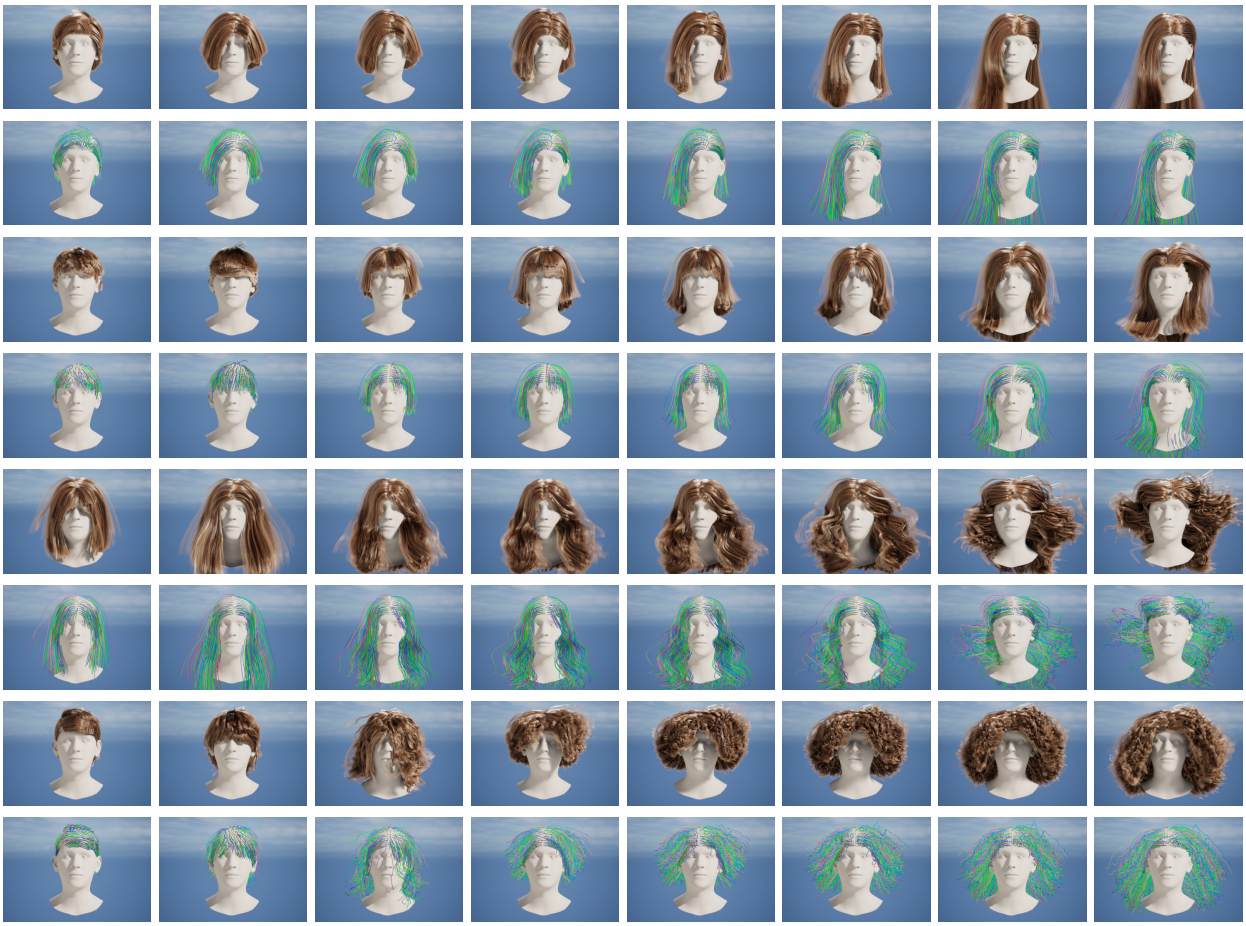

Figure 4. **Classifier-free guidance.** Quality of samples during changing the guidance weight w from 0 to 2.5. Weight $w = 0$ corresponds to unconditional generation, while $w = 1$ - to conditional. For $w > 1$ we obtain over-conditioned results. In our experiments, we fix $w = 1.5$, as higher weights lead to more penetrations and reduced realism. The first four rows correspond to generation samples for the prompt "voluminous straight hair" with two different random seeds, while the last four - for "wavy long hair".

most effective conditioning. To show the upper-bound quality of this metric (denoted as Reference), we calculate the CSIM on our ground-truth dataset with prompts obtained via VQA.

Hairstyle interpolation. We linearly interpolate between two text prompts P_1 and P_2 by conditioning the diffusion model $\mathcal{D}_\theta$ on a linear combination of text embeddings $(1 - \alpha)\tau(P_1) + \alpha\tau(P_2)$, where $\alpha \in [0, 1]$, and τ is the text encoder. For interpolation results obtained for different prompt pairs that differ in length and texture please see Figure 5. One can notice that the interpolation between two types of textures, e.g. *"wavy"* and *"straight"* usually starts appearing for α close to 0.5, while length reduction takes many fewer interpolation steps.

Hairstyle editing. For optimization e_{opt}, we do 1500 steps with the optimizer Adam with a learning rate of 10^{-3}. For diffusion fine-tuning, we do 600 steps with optimizer AdamW [10] with a learning rate of 10^{-4}, $\beta = [0.95, 0.999]$, $\epsilon = 10^{-6}$, and weight decay 10^{-3}. Both stages are optimized using the same reconstruction loss used during the training of the main model. The entire editing pipeline takes around six minutes on a single NVIDIA A100. See Figure 6 for more editing results with and without fine-tuning.

Upsampling scheme. We provide more results on the different upsampling schemes for *"long straight"* and *"long wavy"* hairstyles (see Figure 7). While Blender [1] interpolation in 3D space produces either results with a high level of penetration (bilinear upsampling) or very structured (Nearest Neighbour) hairstyles, we are able to easily blend between two types in latent space, combining the best from the two schemes. Adding noise helps eliminate the grid

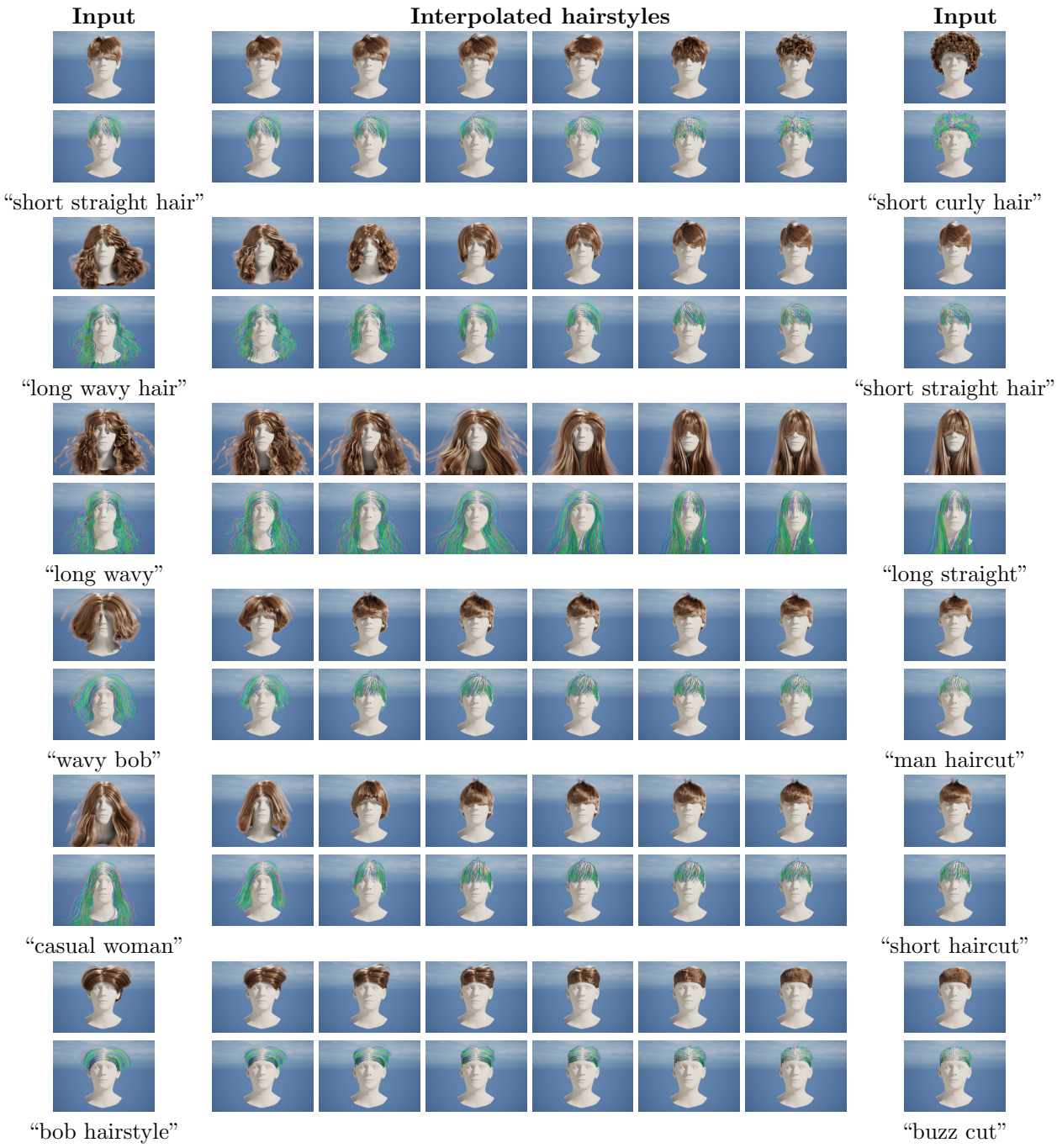

Figure 5. **Hairstyle interpolation.** Linear interpolation between two given textual prompts.

structure inherited from the nearest neighbor sampling and, thus, improves realism. For noising the latent space, we calculate a standard deviation $\widetilde{Z}_\sigma \in \mathbb{R}^{1\times1\times M}$ of latent map after interpolation $\widetilde{Z} \in \mathbb{R}^{N\times N\times M}$, where N is a grid resolution and $M = 64$ is the dimension of latent vector that encodes the entire hair strand. The final noised latent map is $\widetilde{Z} = \widetilde{Z} + \widetilde{Z}_\sigma \odot X \odot Y$, where $X \in \mathbb{R}^{N\times N\times1}$ with elements $x_{ijk} \sim \mathcal{N}(0.15, 0.05)$, $Y \in \mathbb{R}^{N\times N\times1}$ with elements $y_{ijk} = 2q_{ijk} - 1$, where $q_{ijk} \sim \text{Bernoulli}(0.5)$. In such a way, we independently add some small random noise to each latent vector on the mesh grid.

Also, we add results on 3D hairstyle interpolation with our upsampling scheme. In Figure 3 we show our upsampling procedure with blending Bilinear and Nearest

Figure 6. **Hairstyle editing.** Extended editing results of our model. In each section of four images, we provide editing results without additionally tuning the diffusion model (first two rows) and with it (second two rows). Finetuning the diffusion model results in smoother editing and better preservation of input hairstyle.

schemes for strands in 3D space. We found that interpolation in latent space leads to more realistic results.

Generalization capabilities. Our conditional diffusion model can distinguish between different texture types, lengths, bangs, and some popular hairstyles, like the bob, and afro. It models the correlation between gender and hairstyle length, but at the same time, the capacity of the model is limited by the accuracy of the VQA and text encoder system. Asking more general questions improves the generalization quality, but the answers may be less accurate and lead to additional noise during training. To test the generalization capabilities of our model, we evaluate it on out-of-distribution prompts and attempt to generate hairstyles of particular celebrities. We use ChatGPT [11] to describe the hairstyle type of a particular celebrity and use the resulting prompt for conditioning. To our surprise, we find that even given the limited diversity of the hairstyles seen during training, our model can reproduce the general shape of the hairstyle. We show results illustrating the generalization capabilities of our model by reconstructing celebrity hairstyles for *"Cameron Diaz"* and *"Tom Cruise"* (see Figure 8). Between different random seeds hairstyles preserve the main features, like waviness and length, but could change the bang style.

Finally, we show the results of our conditional model on different hairstyle types, by conditioning the model on hairstyle descriptions from [11] (see Figure 9).

Simulations. The hairstyles generated by our diffusion model are further interpolated to resolution 512×512 and then imported into the Unreal Engine [3] as a hair card. We tested simulations in two scenarios: integration into a realistic game environment with manual character control as well as simple rotation movements for different types of hairstyles. The realism of simulations highly depends on the physical characteristics of hair, e.g. friction, stiffness, damping, mass, elasticity, resistance, and collision detection inside the computer graphics engine. An interesting research direction for future work may include the prediction of individual physical properties for each hairstyle that could further simplify the artists' work. For simulation results, please refer to the supplemental video.

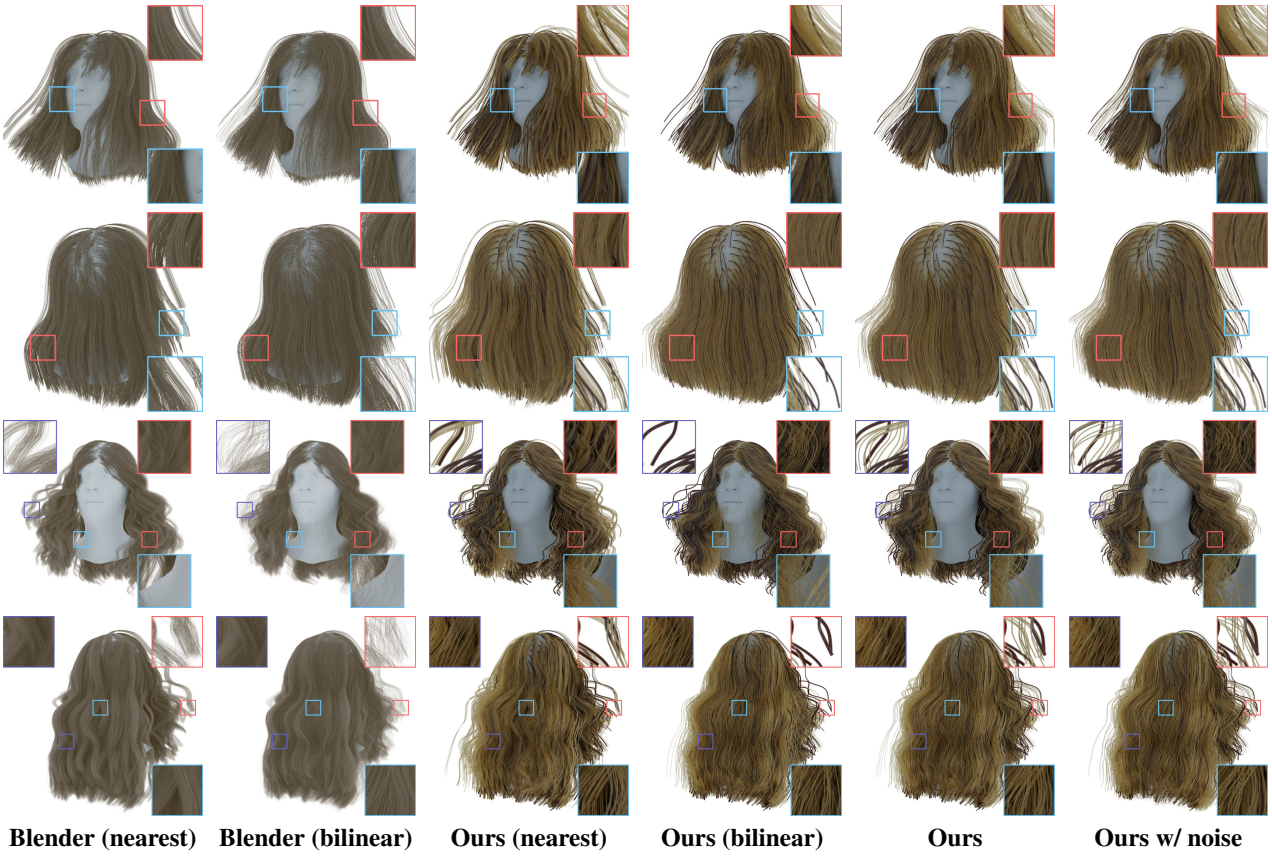

Figure 7. **Upsampling.** Extended results on hairstyle interpolation between guiding strands obtained using different schemes. For better visual comparison, we interpolate hairstyles to around 15,000 strands and additionally visualize guiding strands (shown in dark color) for Ours methods with interpolation in latent space. Our final method with additional noise improves the realism of hairstyles by removing the grid-like artifacts.

Figure 8. **Generalization.** Hairstyles generated for celebrities "Cameron Diaz" (first two rows) and "Tom Cruise" (last two rows) using descriptions from [11]. Several variations of hairstyles with corresponding guiding strands are generated for each celebrity.

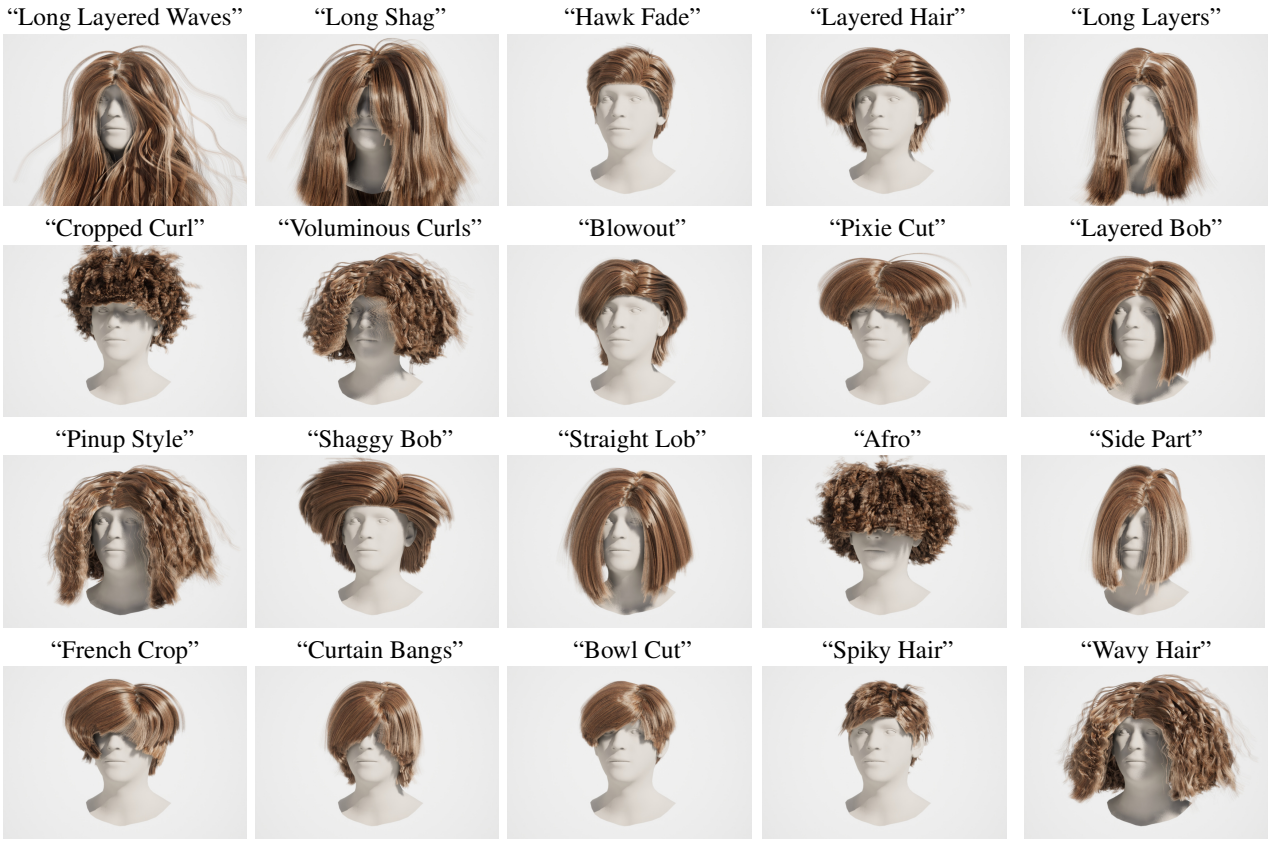

Figure 9. **Conditional generation.** Random samples generated for input prompts with classifier-guidance weight $w = 1.5$.

References

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

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. 1, 2

[3] Epic Games. Unreal engine. 5

[4] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[5] 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. 1

[6] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C.H. Hoi. LAVIS: A one-stop library for language-vision intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 31–41, Toronto, Canada, 2023. Association for Computational Linguistics. 1

[7] 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. 1, 2

[8] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1

[9] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 1, 2

[11] *ChatGPT*. OpenAI, 2023. 4, 5, 6

[12] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1, 2

[13] 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. 1

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 1, 2

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. 1

[16] 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. 1