Instant 3D Human Avatar Generation using Image Diffusion Models
– *Supplementary Material* –

Nikos Kolotouros[1], Thiemo Alldieck[1], Eduard Gabriel Bazavan[1], Enric Corona[1], and Cristian Sminchisescu[1]

Google Research*
{kolotouros,alldieck,egbazavan,enriccorona,sminchisescu}@google.com

1 Implementation Details

1.1 Latent Diffusion models

Our base text-to-image generation model that we finetune is a reimplementation of Stable Diffusion [6] with 800 million parameters trained on internal data sources. The latent space has dimensions $64 \times 64 \times 8$ and the input and output images are $512 \times 512 \times 3$. To enable image conditioning in the models, we pass the conditioning image through the latent encoder, and then we concatenate the conditioning latent with the noisy image latent z_t at the input layer. The weights of the input layer are padded with extra channels initialized with zeros to account for the additional 8 input channels from the conditioning. We train only the parameters of the convolutional layers of the encoder. We finetune the models with the Adam [3] optimizer using a batch size of 64 images and a learning rate of 5×10^{-5} on 16 40GB A100 GPUs, for a total of 40000 training iterations. Finetuning takes around 6 hours. We use both image and text guidance in the style of InstructPix2Pix [2]. We use an text guidance weight of 7.5 and an image guidance weight of 2.0. To enable the use of classifier-free guidance, at training time we randomly drop the conditionings. We mask the input text, input image, or input image and text in a mutual exclusive way, with probability 0.05 each.

1.2 3D Reconstruction model

Our network architecture is similar to PHORHUM [1]. The only modification in the encoder the number of input channels, which is 6 in the case of the front-back model and 12 for the model with the additional GHUM conditioning $\mathcal{G}$. Additionally, because of the ambiguity of lighting estimation, we drop the shading-albedo decomposition and output the shaded color directly. We train our model for 500K iterations with the Adam optimizer using a batch size of 32 and a learning rate of 10^{-4}. Training the model takes 42 hours on 16 40GB A100 GPUs. We use a subset of the original PHORHUM losses for training.

* Now at Google DeepMind.

Average pairwise face similarity $\mathcal{S}$ ↓	
0.64	DreamHuman [4]
0.78	TADA [5]
0.62	Ours
0.55	Random training subjects (lower bound)

Table 1: Generation diversity evaluation. We mark the **best** and second best results. Our method is able to generate faces with larger diversity than the baselines. For reference, we also report the average face similarity for training subjects.

We keep the on-surface loss $\mathcal{L}_g$, inside-outside loss $\mathcal{L}_l$, eikonal loss $\mathcal{L}_e$, and color losses $\mathcal{L}_a$ where we replace the albedo $\boldsymbol{a}$ with the shaded color $\boldsymbol{c}$. We omit the rendering losses $\mathcal{L}_r$ as in our setting we did not find them to be useful and also made training slower.

2 Generation Diversity

In this section we evaluate the diversity of the generations for the different text-to-3D generation methods. We use the same set of 100 generations used in the main paper. As a proxy for diversity we measure the face similarity of the generated subjects. To quantify the face similarity we use the FaceNet embeddings [7]. More specifically, we detect and crop the head regions and then use FaceNet to compute the face embeddings. Given images I_i and I_j with embeddings $\mathbf{e}_i$, $\mathbf{e}_j$ respectively, their pairwise similarity is defined as $s_{ij} = \mathbf{e}_i^T \mathbf{e}_j \in [0, 1]$. The average pairwise similarity $\mathcal{S}$ over a set of images $\mathcal{D}$ is then defined as:

$$\mathcal{S} = \frac{\sum_{I_i \in \mathcal{D}} \sum_{I_j \in \mathcal{D} \setminus \{I_i\}} s_{ij}}{|\mathcal{D}| \cdot (|\mathcal{D}| - 1)}. \tag{1}$$

Intuitively, a high value of $\mathcal{S}$ means that the generated faces are similar to each other. We also consider the maximum pairwise similarity between an image I_i and a reference dataset $\mathcal{D}$ defined as $s_i = \max_{I_j \in \mathcal{D}} s_{ij}$.

The similarity metric s_i quantifies whether the face in image I_i is similar to some face from dataset $\mathcal{D}$.

As shown on Table 1 our method is able to generate more diverse faces than representative optimization-based methods like DreamHuman [4] or TADA [5]. We use the similarity between 100 randomly selected training subjects as reference, and our method is able to generate people with comparable similarity scores.

Additionally, we test whether our method overfits on training identities, by comparing the average maximum similarities of our generations with 200 randomly sampled 3D models from the training and test sets. As reported on Table 2, our generated avatars have the same pairwise similarity scores with either models, thus showing that our model did not overfit on the training set identities.

Average Maximum Similarity	Median Maximum Similarity	
0.67	0.67	Train subjects
0.67	0.67	Test subjects
0.88	0.81	Generations (self-similarity)

Table 2: Evaluating training set memorization. We evaluate the similarity of our generated faces with those from the training and test sets. The results show that our model did not overfit on the training set identities.

3 Additional Qualitative Results

In this section we show additional qualitative results that we could not include in the main paper due to space constraints.

3.1 Relightable Avatar Generation

By design, our image generation models produce images of people with shading. We additionally experiment with generating albedo images instead of shaded ones, and in this way we can create 3D avatars that can then be relighted in different environments. We teach our model to produce albedo images by randomly substituting the shaded model renderings with unshaded ones at training time, and also appending *"uniform lighting"* to the text prompt. In Fig. 2 we show example generations that are rendered in different HDRI environments.

3.2 Semantic editing

We show additional results for the task of semantic editing. We explore 2 scenarios: changing only specific garments on the body while preserving the rest of the appearance and changing the identity of the person wearing the outfit. The input to our method is an image of a person, editing instructions and corresponding semantic segmentation masks. The results are shown on Fig. 1

References

1. Alldieck, T., Zanfir, M., Sminchisescu, C.: Photorealistic monocular 3D reconstruction of humans wearing clothing. In: CVPR (2022) 1
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023) 1
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv (2014) 1
4. Kolotouros, N., Alldieck, T., Zanfir, A., Bazavan, E., Fieraru, M., Sminchisescu, C.: Dreamhuman: Animatable 3d avatars from text. Advances in Neural Information Processing Systems **36** (2024) 2
5. Liao, T., Yi, H., Xiu, Y., Tang, J., Huang, Y., Thies, J., Black, M.J.: Tada! text to animatable digital avatars. In: 3DV (2023) 2
6. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 10684–10695 (2022) 1

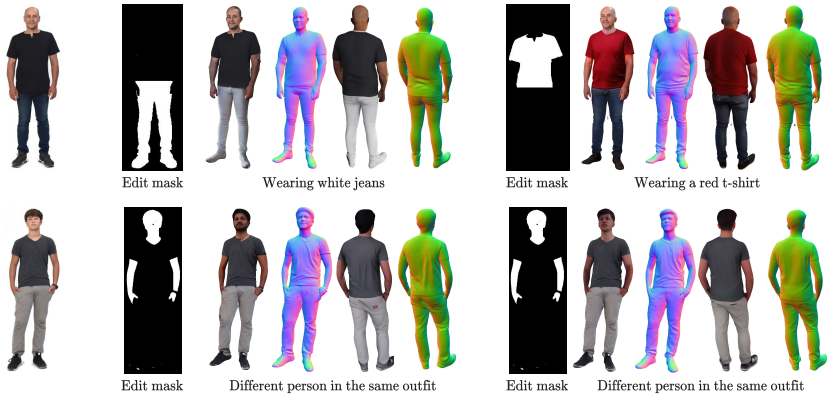

Fig. 1: Semantic editing. In each row we're using the image on the left as input, recover the person's pose and shape parameters and then run our method on updated prompts using the same pose and shape conditioning. In the first row we use additional clothing segmentation masks to perform edits only in specific body regions. In the second row we mask out the person and then generate new people in the same pose wearing the same outfit.

7. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015) 2

Fig. 2: Albedo generation and relighting. The first row shows 8 avatars generated by probing our method to generate albedo instead of shaded colors. The next 3 rows shows the results of rendering the avatars in different HDRI environments.