TECA: Text-Guided Generation and Editing of Compositional 3D Avatars

Hao Zhang[1,3,4*], Yao Feng[1,2*], Peter Kulits[1], Yandong Wen[1]
Justus Thies [1], Michael J. Black[1]
[1]Max Planck Institute for Intelligent Systems, [2]ETH Zürich, [3]Tsinghua University,
[4]RWTH Aachen University, [*]Equal contribution

Project Page: yfeng95.github.io/teca

Figure 1. Compositional avatars: (left) Given a text description, our method produces a 3D avatar consisting of a mesh-based face and body (gray) and NeRF-based style components (e.g. hair and clothing). (middle) Our method generates diverse avatars and hairstyles based on the text input. (right) The non-face parts can be seamlessly transferred to new avatars with different shapes, without additional processing.

Abstract

Our goal is to create a realistic 3D facial avatar with hair and accessories using only a text description. While this challenge has attracted significant recent interest, existing methods either lack realism, produce unrealistic shapes, or do not support editing, such as modifications to the hairstyle. We argue that existing methods are limited because they employ a monolithic modeling approach, using a single representation for the head, face, hair, and accessories. Our observation is that the hair and face, for example, have very different structural qualities that benefit from different representations. Building on this insight, we generate avatars with a compositional model, in which the head, face, and upper body are represented with traditional 3D meshes, and the hair, clothing, and accessories with neural radiance fields (NeRF). The model-based mesh representation provides a strong geometric prior for the face region, improving realism while enabling editing of the person's appearance. By using NeRFs to represent the remaining components, our method is able to model and synthesize parts with complex geometry and appearance, such as curly hair and fluffy scarves. Our novel system synthesizes these high-quality compositional avatars from text descriptions. Specifically, we generate a face image using text, fit a parametric shape model to it, and inpaint texture using diffusion models. Conditioned on the generated face, we sequentially generate style components such as hair or clothing using Score Distillation Sampling (SDS) with guidance from CLIPSeg segmentations. However, this alone is not sufficient to produce avatars with a high degree of realism. Consequently, we introduce a hierarchical approach to refine the non-face regions using a BLIP-based loss combined with SDS. The experimental results demonstrate that our method, Text-guided generation and Editing of Compositional Avatars (TECA), produces avatars that are more realistic than those of recent methods while being editable because of their compositional nature. For example, our TECA enables the seamless transfer of compositional features like hairstyles, scarves, and other accessories between avatars. This capability supports applications such as virtual try-on. The code and generated

avatars will be publicly available for research purposes at yfeng95.github.io/teca.

1. Introduction

There are two traditional approaches to creating facial avatars for games and social media. The first method allows users to select attributes such as skin color, hairstyle, and accessories manually through a graphical interface. While one can create nearly photo-realistic avatars with tools such as MetaHuman [14], the manual process is cumbersome and it is difficult to make an avatar resemble a specific individual's appearance. The second method involves estimating facial shape and appearance from an image [10, 15, 17, 36, 59, 71] or a video [19, 22, 69, 70]. These methods, however, do not support editing and customization of the captured avatar. Recently, a third approach has emerged that exploits text descriptions, generative models, and neural radiance fields [9, 35, 42, 48]. These methods promise the easy creation of diverse and realistic avatars but suffer in terms of 3D realism and editability. To address these challenges, some methods [4, 67] incorporate additional priors from existing face or body models [33, 39] for face generation and animation. However, their ability to synthesize complex geometries like those of hair and scarves is constrained by the fixed topology of typical 3D mesh models.

Going beyond prior work, our goal is to make text a viable interface for creating and editing realistic 3D face avatars with accessories and hair. Our approach is guided by two key observations: 1) Different components of the avatar, such as the hair and face, have unique geometric and appearance properties that benefit from distinct representations. 2) Statistical shape models of head and body shapes can provide valuable guidance to generative image models. To exploit the first observation, we adopt a *compositional* approach to avatar generation, leveraging the strengths of neural and mesh-based 3D content creation methods. Specifically, we model an avatar as a combination of two primary components: the face/body and non-face/body regions. To exploit the second observation, we use the SMPL-X body model [47] to represent the shape of the head and shoulders. By leveraging a model-based representation of face/body shape, we remove the need to model shapes. Instead, we focus such models on creating realistic face texture. Exploiting 3D shapes enables us to generate realistic faces by inpainting textures with existing, pretrained, diffusion models. Moreover, the integration of the shape model enables flexible body shape modifications by manipulating the parametric shape representation, facilitating the transfer of hairstyles and other accessories between avatars with different proportions. For the non-face components like hair, clothing, and accessories, we model their shape and appearance with NeRF [44] since it can represent diverse geometry and reflectance.

Starting with a textual description, our avatar generation process involves multiple steps (see Fig. 2 for an illustration). First, we use stable diffusion model [53] to generate an image of a face conditioned on the text description. Second, we optimize the shape parameters of the SMPL-X body model [47] to obtain a shape representation of the person. Third, inspired by TEXTure [52], we leverage the estimated 3D face shape to rotate the head and use Stable Diffusion to generate missing texture. Fourth, and most important, we employ a sequential, compositional approach to generate additional style components. Conditioned on the face mesh, we learn a NeRF-based component model from the text description. To enable the transfer of non-face components between avatars, we define a canonical space in which we use a template face shape and train the NeRF component on top of it. Using the hybrid volume rendering technique from SCARF [18] and DELTA [19], including shape skinning, we render our hybrid avatar model into the observation image space. We then apply Score Distillation Sampling (SDS) [48] to optimize the NeRF model. The SDS loss provides gradients for updating NeRF to guide 2D renderings to match the text input. While NeRF is a highly flexible method, we argue that it is not needed to represent the face/body. Instead, narrow its focus to modeling specific components of the avatar that are not well represented by parametric face/body models, such as hair or accessories. To that end, we use segmentation to steer the generation. For example, in the case of hair, we compute a hair segmentation mask and use it to focus NeRF on representing the hair region. The segmentation mask is obtained by running CLIPSeg [41] on the current rendering and is updated iteratively throughout the generation process. To enhance learning efficiency, we adopt the Latent-NeRF [42] approach and train the NeRF model in latent space. Finally, we refine the non-face regions using a combination of a BLIP-based loss [31] and SDS in image space. This improves the visual quality of the non-face components.

Figure 1 shows several avatars generated by our method. It shows the underlying body mesh, the generated texture from different views, the generation of NeRF-based hair, hats, and clothing, as well as the transfer of components from one avatar to another. Our approach for avatar generation surpasses existing methods in terms of realism, as demonstrated by extensive qualitative analysis. TECA' compositional framework has two key advantages. First, it uses the "right" models for the task: meshes for the face and body and NeRF for hair and clothing. This disentangling of the face and non-face parts results in avatars of higher realism than the prior art. Second, the compositional nature supports editing of the individual components and enables the seamless transfer of features such as hairstyle or clothing between avatars. These advancements open new possibilities for diverse applications, including virtual try-on.

2. Related work

3D Avatar Creation From X. Creating realistic avatars is a long-standing challenge, with many solutions for building digital avatars from scans, videos, and images. Sophisticated capture systems are used to acquire high-quality 3D scans, which are turned into realistic, personalized, photorealistic avatars [1, 2, 6, 23, 38, 57]. However, these methods are not scalable due to the high cost of building such systems and the sophisticated pipelines required for avatar creation. Consequently, there is great interest in creating avatars from easily captured images [15–17, 59, 71] and monocular videos [8, 19, 20, 22, 69]. Such methods estimate 3D faces with the assistance of parametric models of the head [5, 21, 34] or full body [34, 47]. Recent methods also learn *generative* 3D head models using only images [3, 10], allowing the creation of novel avatars from random noise inputs. Additionally, there exist methods [18, 19, 32, 51, 54] that adopt a compositional approach to avatar modeling, enabling manipulation and control. Unlike these approaches, TECA, requires only natural language descriptions to control the shape, texture, and accessories of a virtual avatar, making the creation and editing of realistic personal avatars accessible to a broad audience.

Text-Guided 3D General Object Generation. Following the success of recent 2D text-to-image models [46, 50, 53], the generation of 3D content from text is gaining attention [27, 35, 42, 48]. Since paired text and 3D training data is scarce, recent text-to-3D methods generate 3D content by leveraging large, pretrained 2D text-to-image models. These approaches learn a 3D representation using losses on the projected image in multiple 2D views, where pretrained models serve as frozen critics. As an example, Contrastive Language–Image Pretraining (CLIP) [49] is used to generate 3D objects in the form of an occupancy network [55], a mesh [13, 43, 45], and a NeRF [27, 61]. Similarly, DreamFusion [48] and subsequent work [12, 26, 35, 42, 58, 64] adopts a text-to-image diffusion model as a guide to optimize 3D object generation, significantly improving visual quality. Despite the rapid progress in generating general objects, these methods suffer from visual artifacts when generating avatars (such as creating a person with multiple faces).

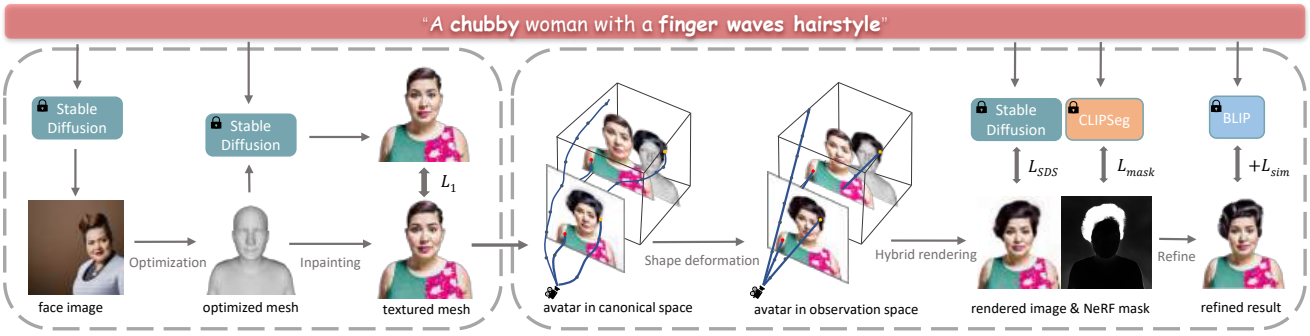
Text-Guided 3D Avatar Generation. Large, pretrained 2D text-to-image models are also used for avatar creation. AvatarCLIP [25] uses the CLIP model to generate coarse body shapes, which are parameterized by the Skinned Multi-Person Linear (SMPL) model [40]. DreamAvatar [9] generates a 3D human avatar from a given text prompt and SMPL body shape, where detailed shape and texture are learned under the guidance of a text-conditioned diffusion model. These methods focus on full-body 3D avatar generation, resulting in low-resolution faces with limited expression. T2P [68] uses CLIP supervision to optimize discrete attributes in a video game character creation engine. As their method is confined to what can be represented by the game engine, their avatars are limited by the expressiveness of the artist-designed hairstyles and facial features. ClipFace [4] enables text-guided editing of a textured 3D morphable face model [34], including expression and texture. Describe3D [66] synthesizes a 3D face mesh with texture from a text description. Neither approach explicitly models hair, and Describe3D resorts to manual post-processing to add hair. To address this problem and boost the visual quality, DreamFace [67] employs a dataset of textures and artist-designed hairstyles, utilizing a CLIP model for component selection and a diffusion model for texture generation. While DreamFace achieves realistic head avatars, it is limited to selecting pre-existing hairstyles from a manually curated gallery of artist-designed assets. Such an approach is expensive and does not scale. Rodin [63] is a generative 3D face model trained from 100K synthetic 3D avatars. They exploit CLIP's text and image embedding to enable text-guided generation and editing. However, their results inherit the limited realism of the synthetic training data, and the approach does not disentangle the hair from the rest of the face. Thus, it is difficult to change hairstyles without unwanted changes in facial appearance. In contrast, our method generates realistic facial avatars using diffusion models without relying on artist-designed hairstyles. Moreover, its compositional nature guarantees that edits to the hairstyle do not impact the face region.

3. Method

TECA generates 3D facial avatars with realistic hair and style components from text descriptions. The overview of the pipeline is illustrated in Fig. 2. Given a text description of a person's physical appearance, we first generate a corresponding image using Stable Diffusion [53]. We extract the 3D geometry by fitting a SMPL-X model to this image. To generate the face texture, we follow the iterative inpainting approach of TEXTure [52], which generates images from different viewpoints and projects them onto the surface. The texture is generated using diffusion [53], taking the text, already-generated texture, and shape into account. Conditioned on the generated face, we generate other components such as hairstyles or clothing using NeRF, which we optimize using SDS constrained with a semantic mask generated with CLIPSeg. The final refinement is done using a combination of SDS and BLIP-based losses.

3.1. Preliminaries

Parametric Body Model. To model a realistic human avatar including the face and shoulders, we use the SMPL-X model [47]. SMPL-X is a parametric mesh model with identity $\boldsymbol{\beta} \in \mathbb{R}^{|\boldsymbol{\beta}|}$, pose $\boldsymbol{\theta} \in \mathbb{R}^{3n_k+3}$, and expression $\boldsymbol{\psi} \in \mathbb{R}^{|\boldsymbol{\psi}|}$ parameters that control the body and facial shape



Face generation (Sec. 3.2) Hair generation (Sec. 3.3)

Figure 2. Overview of TECA. TECA follows a sequential pipeline to generate realistic avatars. First, the text input is passed to Stable Diffusion to generate a single face image, which serves as a reference to obtain the geometry by SMPL-X fitting. We then adopt a texture painting approach inspired by [52], where the mesh is iteratively painted with a texture corresponding to the text using Stable Diffusion. Subsequently, style components such as the hair are modeled using NeRF in a latent space, with optimization guided by an SDS loss (L_{SDS}) and a mask loss (L_{mask}) with CLIPSeg segmentation, and finally refined in pixel space using L_{SDS}, L_{mask}, and an additional BLIP-based loss (L_{sim}). This hybrid modeling approach results in high-quality and realistic avatars.

of the avatar. SMPL-X provides a consistent, predefined topology with a set of vertices $\boldsymbol{V} \in \mathbb{R}^{n_v \times 3}$, which are computed by:

$$\begin{aligned} \boldsymbol{V} &= F_{\text{smplx}}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) \\ &= F_{\text{lbs}}(T_P(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}), J(\boldsymbol{\beta}), \boldsymbol{\theta}; \boldsymbol{W}) \end{aligned} \tag{1}$$

where F_{lbs} is a linear blend skinning function and $\boldsymbol{W} \in \mathbb{R}^{n_k \times n_v}$ are the blend weights. T_P represents the template mesh in a neutral, canonical pose:

$$T_P(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) = \boldsymbol{T} + B(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}), \tag{2}$$

where $B(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) : \mathbb{R}^{|\boldsymbol{\beta}|} \times \mathbb{R}^{3n_k+3} \times \mathbb{R}^{|\boldsymbol{\psi}|} \rightarrow \mathbb{R}^{n_v \times 3}$ gives deformations of the template based on the shape, pose, and expression parameters. $J(\boldsymbol{\beta}) : \mathbb{R}^{|\boldsymbol{\beta}|} \rightarrow \mathbb{R}^{n_k \times 3}$ is a joint regressor that takes in the identity shape parameters and produces the positions of the body joints. SMPL-X builds a connection between the mean body shape $\boldsymbol{T}$ and a specific body shape $\boldsymbol{V}$ with identity, pose, and expression information. This can be formulated as a vertex-wise mapping from $\boldsymbol{T}$ to $\boldsymbol{V}$. Specifically, given $\boldsymbol{t}_i$ and $\boldsymbol{v}_i$ (the i-th row from $\boldsymbol{T}$ and $\boldsymbol{V}$, respectively), the mapping is:

$$\begin{bmatrix} \boldsymbol{v}_i^T \\ 1 \end{bmatrix} = M_i(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) \begin{bmatrix} \boldsymbol{t}_i^T \\ 1 \end{bmatrix}. \tag{3}$$

Here, the function $M_i(\cdot)$ produces a 4×4 matrix, $M_i(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) =$

$$\left(\sum_{k=1}^{n_k} w_{k,i} G_k(\boldsymbol{\theta}, J(\boldsymbol{\beta})) \right) \begin{bmatrix} \boldsymbol{E} & B_i(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi})^T \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{4}$$

where $w_{k,i}$ is an entry from the blend skinning weights $\boldsymbol{W}$ and $G_k(\boldsymbol{\theta}, J(\boldsymbol{\beta})) \in \mathbb{R}^{4 \times 4}$ computes the world transformation for the k-th body joint. $\boldsymbol{E} \in \mathbb{R}^{3 \times 3}$ is the identity matrix, and B_i computes the i-th row of the blend shapes.

NeRF Representation. NeRF [44] encodes a 3D object as a continuous volumetric radiance field of color $\boldsymbol{c} \in \mathbb{R}^{|\boldsymbol{c}|}$ and density $\sigma \in \mathbb{R}$. A NeRF is represented by a neural network $(\sigma, \boldsymbol{c}) = F_{\text{nerf}}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{\Phi})$, where $\boldsymbol{x} \in \mathbb{R}^3$ is the location, $\boldsymbol{p} \in \mathbb{R}^2$ is the viewing direction, and $\boldsymbol{\Phi}$ are the learnable parameters of F_{nerf}. Given a camera position, we estimate a 2D image from the NeRF with volume rendering. We denote a per-pixel ray $R(\ell) = \boldsymbol{o} + \ell \boldsymbol{d}$ by the origin $\boldsymbol{o} \in \mathbb{R}^3$, direction $\boldsymbol{d} \in \mathbb{R}^3$, and $\ell \in [\ell_n, \ell_f]$. To discretize the rendering, we evenly split the rendering range into n_ℓ bins and randomly sample a ℓ_i for every bin with stratified sampling. The volume rendering formulation for each pixel is:

$$\begin{aligned} C(R(\ell)) &= \sum_{i=1}^{n_\ell} \alpha_i \boldsymbol{c}_i, \\ \text{with } \alpha_i &= \exp\left(- \sum_{j=1}^{i-1} \sigma_j \Delta\ell_j \right) \left(1 - \exp\left(-\sigma_i \Delta\ell_i\right)\right). \end{aligned} \tag{5}$$

Here, $\Delta\ell_i = \ell_{i+1} - \ell_i$ is the adjacent samples distance.

Score Distillation Sampling. DreamFusion [48] proposes Score Distillation Sampling (SDS) to guide 3D content generation using pre-trained 2D text-to-image diffusion models. Following [48], we denote the learned denoising function in the diffusion model as $\epsilon(\boldsymbol{Q}_t, \boldsymbol{y}, t)$. Here, $\boldsymbol{Q}_t$ is the noisy image at timestep t. SDS adopts the denoising function as a critic to update the 2D rendering $\boldsymbol{Q}$ of the generated 3D object across different viewpoints. The gradient is computed as:

$$\nabla_{\boldsymbol{Q}} L_{\text{sds}}(\boldsymbol{Q}) = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[\boldsymbol{u}_t \cdot \left(\epsilon\left(\boldsymbol{Q}_t, \boldsymbol{y}, t\right) - \boldsymbol{\epsilon} \right) \right], \tag{6}$$

where $\boldsymbol{u}_t$ is a weight at timestep t [24].

3.2. 3D Face Generation

To generate a 3D facial shape from a text description, we use a pre-trained Stable Diffusion model [53] to synthesize

a 2D face image that semantically matches the given text. The descriptor keywords might include overweight, slim, muscular, or old. Given the generated face image, we use an off-the-shelf landmark detector [7] to obtain a set of facial landmarks $\{\boldsymbol{e}_i | \boldsymbol{e}_i \in \mathbb{R}^3\}_{i=1}^{n_e}$, where n_e is the number of landmarks. We optimize the SMPL-X parameters using:

$$(\boldsymbol{\beta}^*, \boldsymbol{\theta}^*, \boldsymbol{\psi}^*) = \underset{\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{\psi}}{\operatorname{argmin}} \sum_{i=1}^{n_e} \left\| M_{\kappa(i)}(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi}) \boldsymbol{t}_{\kappa(i)} - \boldsymbol{e}_i \right\|_1 \quad (7)$$

$\kappa(i)$ denotes the index of a vertex of the SMPL-X model that corresponds to the i-th landmark. Note that optimizing facial shape in this way results in full-body shape parameters, but the pose parameters are not well-constrained. Then, the final avatar shape is given by $\boldsymbol{V}^* = F_{\text{smplx}}(\boldsymbol{\beta}^*, \boldsymbol{\theta}^c, \boldsymbol{0})$, where $\boldsymbol{\theta}^c$ represents the body pose parameters corresponding to an "A-pose".

To generate a complete texture of the reconstructed face, we follow the iterative inpainting procedure of TEXTure [52]. In each step, we generate an image $\boldsymbol{I}_i$ in the viewpoint of $\boldsymbol{p}_i$ using Stable Diffusion [53] and project $\boldsymbol{I}_i$ back to the mesh surface according to the geometry $\boldsymbol{V}^*$ and $\boldsymbol{p}_i$. The iterative inpainting can be denoted $\{\boldsymbol{A}_i\}_{i=0}^{n_p}$, where $\boldsymbol{A}_0$, $\{\boldsymbol{A}_i\}_{i=1}^{n_p-1}$, and $\boldsymbol{A}_{n_p}$ are the initial, intermediate, and final texture UV maps, respectively. Denoting the differentiable mesh renderer as $R_m(\boldsymbol{A}, \boldsymbol{p}, \boldsymbol{V}^*)$, the painting process can be summarized as:

$$\boldsymbol{A}_i = \operatorname{argmin}_{\boldsymbol{A}} \left\| R_m(\boldsymbol{A}, \boldsymbol{p}_i, \boldsymbol{V}^*) - \boldsymbol{I}_i \right\|_1 . \quad (8)$$

To reduce cross-view conflicts in texture, we follow [52] to take both the mesh geometry $\boldsymbol{V}^*$ and previous texture $\boldsymbol{A}_{i-1}$ information into account when generating the image $\boldsymbol{I}_i$. This is achieved by iteratively applying depth-aware and inpainting diffusion models in the denoising process. We also make the assumption that the face of the individual is approximately bilaterally symmetric and add an additional symmetry regularization term L_{sym}. This term enforces similarity between the frontal face image and its horizontally flipped counterpart. Further information on this regularization and our texture generation process can be found in the Sup. Mat.

3.3. Hair, Clothing, and Accessory Generation

Canonicalization. Building upon the generated face, represented as a textured mesh, we learn a separate NeRF model for attributes like hair, clothing, or accessories. The NeRF model is built in a canonical space, which is constructed around the SMPL-X template mesh $\boldsymbol{T}$, enabling the animation and transfer of the non-face parts. For body mesh $\boldsymbol{V}$ and its corresponding parameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and $\boldsymbol{\psi}$, we follow previous work [9, 11, 18] to map the points from observation space to canonical space ($\boldsymbol{x} \rightarrow \boldsymbol{x}^c$):

$$\boldsymbol{x}^c = \sum_{\boldsymbol{v}_i \in \mathcal{N}(\boldsymbol{x})} \frac{\omega_i(\boldsymbol{x}, \boldsymbol{v}_i)}{\omega(\boldsymbol{x}, \boldsymbol{v}_i)} M_i(\boldsymbol{0}, \boldsymbol{\theta}^c, \boldsymbol{0}) \left(M_i(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\psi})\right)^{-1} \boldsymbol{x}, \quad (9)$$

where $\mathcal{N}(\boldsymbol{x})$ is the set of nearest neighbors of $\boldsymbol{x}$ in $\boldsymbol{V}$. The weights are computed as:

$$\begin{aligned} \omega_i(\boldsymbol{x}, \boldsymbol{v}_i) &= \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{v}_i\|_2 \left\| \boldsymbol{w}_{\xi(\boldsymbol{x})} - \boldsymbol{w}_i \right\|_2}{2\tau^2} \right), \text{ and} \\ \omega(\boldsymbol{x}, \boldsymbol{v}_i) &= \sum_{\boldsymbol{v}_i \in \mathcal{N}(\boldsymbol{x})} \omega_i(\boldsymbol{x}), \end{aligned} \quad (10)$$

where $\xi(\boldsymbol{x})$ is the index of the vertex in $\boldsymbol{V}$ that is closest to $\boldsymbol{x}$. $\boldsymbol{w}_i$ is the i-th column of $\boldsymbol{W}$, and τ is 0.2.

Mesh-Integrated Volumetric Rendering. Conditioned on the textured face mesh, we learn NeRF models with mesh-integrated volume rendering following [18]. Specifically, when a ray $R(\ell)$ is emitted from the camera center $\boldsymbol{o}$ and intersects the mesh surface, we set ℓ_f such that $R(\ell_f)$ represents the first intersection point. The texture color at $R(\ell_f)$, denoted by $\boldsymbol{c}^*$, is then used in the volume rendering by extending Eqn. 5:

$$\begin{aligned} C(R(\ell)) &= \left(1 - \sum_{i=1}^{n_\ell - 1} \alpha_i\right) \boldsymbol{c}^* + \sum_{i=1}^{n_\ell - 1} \alpha_i \boldsymbol{c}_i, \\ \text{with } \alpha_i &= \exp\left(-\sum_{j=1}^{i-1} \sigma_j \Delta\ell_j\right) \left(1 - \exp(-\sigma_i \Delta\ell_i)\right). \end{aligned} \quad (11)$$

For a ray that does not intersect the mesh surface, the computation of aggregated color follows Eqn. 5. Unlike in a traditional NeRF representation, we model NeRF in a latent space to accelerate the learning process. Specifically, $\boldsymbol{c}$ and $\boldsymbol{c}^*$ represent 4-dimensional latent features. While the features $\boldsymbol{c}_i$ are optimized, the latent feature on the mesh surface $\boldsymbol{c}^*$ is obtained by running the stable diffusion encoder [53] on the RGB rendering of the textured mesh. After volumetric integration, the resulting feature image is decoded with the Stable Diffusion model.

To train a NeRF model for a specific style component, we rely on CLIPSeg [41] for spatial guidance. Taking hair as an example, we use CLIPSeg with a keyword of "hair" to segment the hair region in the image generated by rendering the hybrid mesh–NeRF model. This hair segmentation mask $\boldsymbol{\Omega}$ indicates the hair and non-hair regions. We use $\boldsymbol{\Omega}$ to guide the NeRF model to focus on the representation of objects within the masked region while discouraging it from learning geometry outside the region. This is useful to prevent the NeRF from modeling the face, when it should only represent hair. We use the following mask-based loss:

$$L_{\text{mask}} = \left\| \boldsymbol{\Omega} - \hat{\boldsymbol{\Omega}} \right\|_1, \quad (12)$$

where $\hat{\boldsymbol{\Omega}}$ is the rendered NeRF mask, obtained by sampling rays for all pixels of the entire image. The computation of a mask value at pixel location $R(\ell)$ is given by:

$$\Omega_i\left(R(\ell), n_\ell - 1\right) = \sum_{i=1}^{n_\ell - 1} \alpha_i. \quad (13)$$

Figure 3. Qualitative comparison with SOTA methods. Text prompts: (a) "An overweight man with a crown-braid hairstyle," (b) "A man in a purple hoodie," (c) "An old bald African woman wearing a wool scarf," (d) "A young woman in a military hat."

To prevent floating "radiance clouds," we incorporate the sparsity loss from [42] into the training of the NeRF:

$$L_{\text{sparse}} = \sum_i F_{\text{BE}}\left(\Omega_i\left(R(\ell), n_\ell\right)\right), \tag{14}$$

where $F_{\text{BE}}(a) = -a\ln a - (1-a)\ln(1-a)$ is a binary entropy function. The total training loss function for the latent NeRF models is:

$$L_{\text{NeRF}} = L_{\text{sds}} + \lambda_{\text{mask}} L_{\text{mask}} + \lambda_{\text{sparse}} L_{\text{sparse}}. \tag{15}$$

Style Component Refinement in RGB Space. While latent NeRF models can be used to learn reasonable geometry and texture, we have observed that adding further refinement in RGB space using a combination of the SDS loss and a loss based on BLIP [31] improves local detail. Our BLIP loss, L_{sim}, measures the similarity of high-level visual and text features. Maximizing their similarity encourages the NeRF model to capture additional details, including structure, texture, and semantic content from the text description, leading to visually appealing results, see Fig. 7.

To perform the refinement, we append an additional linear layer to the NeRF model that converts the 4D latent feature into a 3D color representation [60]. The initial weights of this layer are computed using pairs of RGB images and their corresponding latent codes over a collection of natural images. Let $\boldsymbol{z}_{\text{img}}$ and $\boldsymbol{z}_{\text{text}}$ be the embeddings of the rendered image and text prompt, then the similarity loss is:

$$L_{\text{sim}} = -\frac{\boldsymbol{z}_{\text{img}}^T \boldsymbol{z}_{\text{text}}}{\|\boldsymbol{z}_{\text{img}}\| \cdot \|\boldsymbol{z}_{\text{text}}\|}, \tag{16}$$

and the learning objective in refinement stage is:

$$L_{\text{refine}} = L_{\text{NeRF}} + \lambda_{\text{sim}} L_{\text{sim}}. \tag{17}$$

More implementation details are included in Sup. Mat.

4. Experiments

We evaluate TECA through 1) comparisons with state-of-the-art (SOTA) methods for text-guided generation, 2) an online perceptual study, 3) quantitative evaluation, 4) the application of try-on and animation of generated avatars, and 5) ablation studies exploring our design choices. To evaluate TECA's ability to generate diverse compositional avatars, we need text prompts that are also compositional. These text prompts may include facial attributes (*e.g.* overweight, slim, muscular), hairstyles (*e.g.* bun, afro, braid), clothing (*e.g.* jacket, wool scarf, hat), and more. The candidate attributes and styles words are taken from a dataset of faces [37], a dataset of hairstyles [65], and other online sources[1][2]. In total, these attributes produce up to 3,300 text-prompt combinations, ensuring rich diversity.

[1] vocabulary-clothing-clothes-accessories
[2] types-of-hats

4.1. Comparisons with SOTA Methods

We compare TECA with four SOTA methods. Two are solely based on NeRF representations (SJC [62] and Latent-NeRF [42]) and two are based on mesh painting techniques (Latent-Paint [42] and TEXTure [52]). Figure 3 shows generated examples obtained from four diverse text prompts describing various personal characteristics, hairstyles, and clothing. Notably, all methods successfully generate avatars with recognizable features that semantically align with the text, such as gender, color, and clothing items. However, SJC and Latent-NeRF produce visually distorted and incomplete avatars, primarily due to flawed geometry and low-resolution textures. While Latent-Paint incorporates a mesh as a shape prior, leading to reasonable proportions, the textures still suffer from blurriness and a lack of cross-view consistency. TEXTure demonstrates good texture quality but is limited by the mesh topology; it cannot non-body components like the crown-braid hairstyle and hoodie. In contrast, TECA generates more realistic and natural avatars with strong cross-view consistency. Our text-generated avatars exhibit detailed appearances, including diverse hairstyles and accessories (see Sup. Mat.).

4.2. Perceptual Study

We conducted an online perceptual study to rate avatars synthesized by TECA and the baseline methods, SJC, Latent-NeRF, LatentPaint, and TEXTure. Participants on Amazon Mechanical Turk were presented with videos of the generated avatars and asked to rate the visual realism and consistency with the text prompt using a seven-point Likert scale. We showed each participant the same thirty prompts but shuffled which method's avatar the participants saw for each prompt. Each participant was shown an equal number of avatars synthesized by each method. A total of 150 responses were collected, out of which 52 (35%) participants passed all of the catch trials and were included in the study. We applied the non-parametric Friedman test and then performed the Nemenyi post-hoc test to identify pairwise differences. The results are shown in Fig. 4 and only our method receives, on average, positive ratings to both questions. See Sup. Mat. for more details.

4.3. Quantitative Evaluation

We also quantitatively compare our method with other state-of-the-art methods. Specifically, we assess the semantic matching between text and avatar using the CLIP [49] score and evaluate the generated avatar quality through the Fréchet Inception Distance (FID) [56]. To compute the CLIP score, we convert the videos used in Section 4.2 into images and employed CLIP [49] to compute the cosine distance between these images and their respective text prompts. The results are shown in Table 1. Our method achieves the highest semantic consistency, indicating its ability to accurately represent text descriptions. For FID, we generated 200 avatars using 40 different text prompts, with 5 random seeds for each prompt. Each avatar was rendered from 50 different views, resulting in a total of 10,000 images for evaluation. The ground truth distribution was based on the first 10,000 images from the Flickr-Faces-HQ Dataset (FFHQ) [28]. The FID scores are shown in Table 1. TECA has the lowest FID score, indicating its superior image quality relative to the other methods.

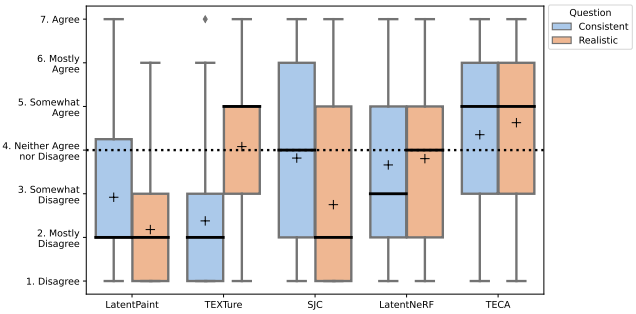


Figure 4. A box-and-whisker plot of the perceptual study results. Users were asked the questions 1) "The appearance of the avatar in the video matches the text description below it." (blue color) and 2) "The avatar in the video is visually realistic" (orange color), '+' corresponds to the mean.

Figure 5. TECA Try-On. The hairstyle, hat, scarf, or clothing is transferred from the source avatars to the target avatar.

Method	CLIP Score ↑	FID↓
SJC [62]	0.2979	27.50
Latent-NeRF [42]	0.3025	24.49
Latent-Paint [42]	0.2854	38.33
TEXTure [52]	0.2761	29.51
Ours	**0.3213**	**14.98**

Table 1. Quantitative evaluation results. Higher CLIP score indicates better consistency between the text prompts and generated avatars, lower FID indicates higher realism of the avatars.

4.4. Applications: Try-on and Animation

Since TECA is compositional, components like hairstyles and accessories can be transferred between avatars. As shown in Fig. 5, TECA can transfer a brown pullover, crown-braid hairstyle, red scarf, or blue hat to the target

Figure 6. TECA Animation. The generated avatar is animated to different poses and expressions.

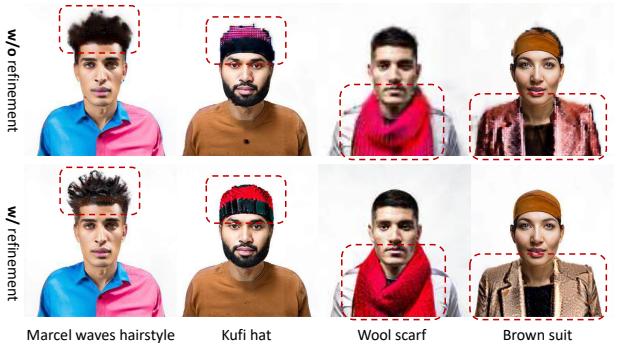


Figure 7. Comparison of results between unrefined (top) and refined (bottom) style components. The refinement improves the details of hair, hat, scarf, and clothing. The refined components are indicated by red dotted line boxes.

avatar. The non-face components adjust in size to adapt to a new head shape. This makes our generated hairstyles and accessories highly versatile. For instance, users can input their own avatars and transfer a learned hairstyle to it.

Leveraging the SMPL-X model, we also gain the flexibility to animate the avatar across various poses and expressions. As detailed in Sec. 3.3, the NeRF component has the capacity to synchronize its movement with the dynamics of the SMPL-X body. Displayed in Fig. 6, our generated avatar can be animated with varying head poses and expressions. Notice the transition from a neutral to an open-mouth expression (middle image), where the interior mouth region is absent. This limitation could be addressed by inpainting the inside mouth region using diffusion models. Further instances showcasing this including face editing results are available in the Sup. Mat.

4.5. Ablation Experiments

Non-Face Refinement. We investigate the effect of refining non-face details using a combination of SDS and BLIP losses. Figure 7 illustrates the difference between refined and non-refined style components. The results demonstrate that the refinement produces more detail, noticeably enhancing the overall visual quality.

CLIPSeg Segmentation. The segmentation loss prevents NeRF from trying to represent the entire avatar and focuses it on representing a specific part. Without the loss, the results are significantly worse; see Fig. 8.

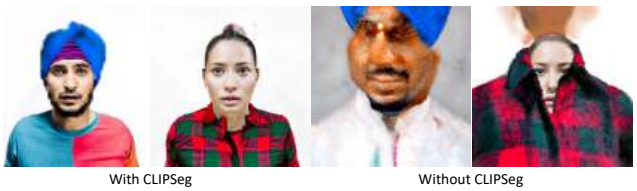


Figure 8. Ablation of CLIPSeg. Without the segmentation information, NeRF learns the entire avatar instead of the components.

Figure 9. Failure cases showing the impact of poor segmentation.

5. Discussion and Limitations

Segmentation with CLIPSeg. To guide the learning process, we use CLIPSeg to obtain masks for the region of interest, such as hair or clothing. This encourages the NeRF to focus on learning specific components rather than on the entire avatar. Our method's effectiveness is contingent on the segmentation quality. If CLIPSeg encounters challenges, flawed NeRF representations may result, such as floating points in the face region, as shown in Fig. 9.

Performance of Diffusion Models. Our results are constrained by the capabilities and biases of pretrained diffusion models because they provide the semantic information for avatar generation.

Dynamics. TECA's ability to animate avatars via the SMPL-X parametric model highlights its potential, yet addressing complex dynamics in elements like hair and clothing calls for further exploration.

Relighting. Our model does not support relighting in new environments, as the learned RGB color for the face texture and NeRF-based hair or accessories is baked with lighting. Further work is needed to disentangle albedo and lighting attributes to enable relighting.

6. Conclusion

We presented TECA, an innovative method for generating realistic 3D facial avatars with hair and accessories from text descriptions. By adopting a compositional model and using distinct representations for different components, we addressed the limitations of existing methods in terms of realism, shape fidelity, and capabilities for editing. Our experimental results demonstrate the superior performance of TECA compared to state of the art, delivering highly detailed and editable avatars. Further, we demonstrated the transfer of hairstyles and accessories between avatars.

Disclosure This work was partially supported by the Max Planck ETH Center for Learning Systems. MJB has received research gift funds from Adobe, Intel, Nvidia, Meta/Facebook, and Amazon. MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH. While MJB is a consultant for Meshcapade, his research in this project was performed solely at, and funded solely by, the Max Planck Society.

7. Appendix

7.1. Implementation Details

For the SMPL-X model [47], we use $|\boldsymbol{\beta}| = 300$ and $|\boldsymbol{\psi}| = 100$, and we use 68 facial landmarks to fit the SMPL-X shape to the reference images, ($n_e = 68$). In the SMPL-X optimization process, we incorporate shape and expression regularization [47] with a weight of $5e-5$. For texture generation on the mesh, we use $n_p = 10$ viewing directions. The mesh texture is optimized in an iterative manner from various view angles, following the sequence illustrated in Fig. 10. The symmetry loss L_{sym} is applied during this process, leveraging our assumption that the face is approximately symmetric. Specifically, for front and back views (No. 1 and No. 10), we apply an L_2 loss between the rendered image $R_m(\boldsymbol{A}, \boldsymbol{p}_i, \boldsymbol{V}^*)$ that renders the geometry $\boldsymbol{V}^*$ in the view direction $\boldsymbol{p}_i$ based on the UV map $\boldsymbol{A}$ and the horizontally flipped version of the Stable Diffusion [53] generated image $\boldsymbol{I}'_i$. For right views (Nos. 3, 5, 7, and 9), we implement the L_2 loss by comparing the rendered images and the corresponding left views (Nos. 2, 4, 6, and 8). During NeRF training, we sample 96 points each ray ($n_\ell = 96$). During the refinement stage, we increase the number of sampled points n_ℓ to 128. In the optimization, we employ a latent code with dimensions of $64 \times 64 \times 4$. During refinement, we render the image at a resolution of $480 \times 480 \times 3$. The network used in the NeRF training process comprises three fully connected layers, and the adapter from the latent space to the pixel space is a one-layer MLP following [60]. Other hyperparameters include $\tau = 0.1$, $|\mathcal{N}(\boldsymbol{x})| = 6$, $\ell_n = -1$, and $\ell_f = 1$.

For the Stable Diffusion model [53] and its depth-aware and inpainting variants, we use the implementations available on HuggingFace[3][4][5]. For the BLIP model [31], we use the released model[6] from the LAVIS project [30]. We employ the Adam optimizer [29] with a learning rate of 0.01 for texture optimization, while for other optimizations, we use a learning rate of 0.001. For loss weights, we fix $\lambda_{\text{sym}} = 0.5$, $\lambda_{\text{mask}} = 0.1$, $\lambda_{\text{sparse}} = 0.0005$, and $\lambda_{\text{sim}} = 1$. The average run time for avatar generation is currently around three hours on an A100 GPU.

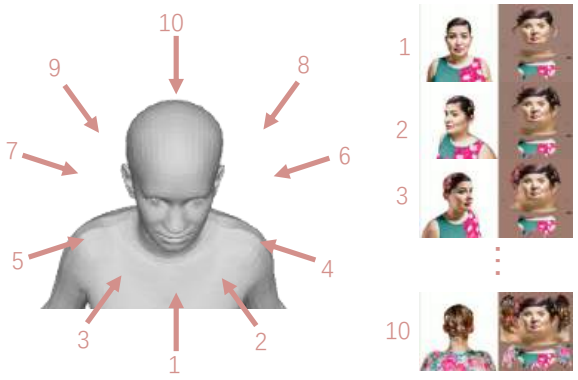


Figure 10. Left: The ten virtual camera views used for the texture optimization process. Right: The rendered images (1st column) and the corresponding UV maps (2nd column).

7.2. Perceptual Study Details

In our survey, we used a random selection process to create a set of thirty unique prompt combinations, each consisting of five elements. These elements were: 1) gender (male or female); 2) a color[7]; 3) a hairstyle[8] or hat[9], weighted equally; 4) another color; and 5) a type of upper-body clothing[10]. These combinations were used to construct a prompt in the form of "A [1] with a(n) [2] [3] wearing a(n) [4] [5]".

To mitigate potential interaction effects resulting from participants' unfamiliarity with the styles presented, we included an image from the internet that represented the hairstyle or type of hat described in the prompt; the image was displayed next to the avatar video. Participants were then asked to rate their agreement with two statements: 1) "The avatar in the video is visually realistic" and 2) "The appearance of the avatar in the video matches the text description below it." To determine whether a participant successfully passed the catch trials, we examined their ratings for both questions. Participants were considered to have passed if they rated both questions with greater-than-neutral agreement or greater-than-neutral disagreement on all five constant manually curated high-quality samples and catastrophic generation failures, respectively. A total of 150 responses were collected, out of which 52 (35%) participants passed all of the catch trials and were included in the study. The response distributions failed the Shapiro–Wilk normality test, so we applied the non-parametric Friedman test, which indicated that the method used to generate the avatar had a statistically significant effect on the outcomes of both study questions. Subsequently, we performed the Nemenyi post-hoc test to identify pairwise differences. Us-

[3] stabilityai/stable-diffusion-2
[4] stabilityai/stable-diffusion-2-depth
[5] stabilityai/stable-diffusion-2-inpainting
[6] BLIP/models/model_base.pth

[7] HairCLIP/main/README.md
[8] HairCLIP/main/mapper/hairstyle_list.txt
[9] 7esl.com/types-of-hats
[10] 7esl.com/vocabulary-clothing-clothes-accessories

Figure 11. Additional examples for generated avatars by our method.

Figure 12. Additional examples of accessories such as earrings, necklaces, and glasses.

ing a significance level (α) of 0.05, the perceived *realism* of TECA was determined to be significantly different than that of all baselines other than TEXTure, and the *text consistency* was determined to be significantly different than that of all baselines. These findings confirm our initial expectations regarding the strengths of each method and support the value of our proposed combination of mesh-based and NeRF-based representations.

7.3. More Qualitative Results

More generation results. Avatars with distinct hairstyles, clothing, and accessories are shown in Fig. 11. In addition, Fig. 15 shows instances with a variety of other types of accessories, such as earrings, necklaces, and glasses. To illustrate the diversity of the generated avatars, Fig. 13 shows avatars generated using the same prompt but different seeds, producing diverse faces and clothing that consistently align with the text input. Fig. 14 presents avatars that are generated using more detailed descriptions, highlighting the alignment between the avatars and the input texts.

More applications. TECA allows for the transfer of hairstyles and accessories, as demonstrated by additional examples in Fig. 15. Moreover, the generated avatars can be animated with different expressions. As our method automatically estimates the SMPL-X shape parameters of this subject, we can then change the expression parameters of SMPL-X model to animate the face. Regarding face texture, the inner mouth region is missing. To address this, we apply an inpainting Stable Diffusion model to inpaint the missing area. The results are shown in Fig. 16. We can further edit the avatar using text input. For example, Fig. 17 shows the results of altering the color of lips and eyes.

Figure 13. Additional examples of the diversity of avatars generated with the same text prompt: “a woman with long hair wearing a shirt.”

Figure 14. Additional examples of generated avatars with more detailed text descriptions such as colors and detailed shapes.

Figure 15. Additional examples for hair and accessory transfer.

Figure 16. Additional examples avatars generated with different expressions. The in-mouth area is represented with a painted texture. The expressions from top to bottom are: smiling, angry, disgusted, and screaming.

Figure 17. Additional examples of editing the generated avatar's texture. The first line changes the lip color with the "red lip" prompt and the second line alters the eye color with prompts "green eyes" and "brown eyes".

References

[1] Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul Debevec. The Digital Emily project: Achieving a photorealistic digital actor. *IEEE Computer Graphics and Applications*, 30 (4):20–31, 2010. 3

[2] Oleg Alexander, Graham Fyffe, Jay Busch, Xueming Yu, Ryosuke Ichikari, Andrew Jones, Paul Debevec, Jorge Jimenez, Etienne Danvoye, Bernardo Antionazzi, Mike Eheler, Zybnek Kysela, and Javier von der Pahlen. Digital Ira: Creating a real-time photoreal digital actor. In *ACM SIGGRAPH 2013 Posters*, New York, NY, USA, 2013. Association for Computing Machinery. 3

[3] Sizhe An, Hongyi Xu, Yichun Shi, Guoxian Song, Umit Ogras, and Linjie Luo. Panohead: Geometry-aware 3d full-head synthesis in 360°, 2023. 3

[4] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. Clipface: Text-guided editing of textured 3d morphable models, 2023. 2, 3

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Transactions on Graphics, (Proc. SIGGRAPH)*, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 3

[6] George Borshukov and J. P. Lewis. Realistic human face rendering for "the matrix reloaded". In *ACM SIGGRAPH 2005 Courses*, page 13–es, New York, NY, USA, 2005. Association for Computing Machinery. 3

[7] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1021–1030, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 5

[8] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, et al. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 3

[9] 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, 2023. 2, 3, 5

[10] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16133, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2, 3

[11] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular rgb videos, 2021. 5

[12] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation, 2023. 3

[13] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition, 2022. 3

[14] Epic Games. Metahuman, 2023. 2

[15] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018. 2, 3

[16] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael Black. Collaborative regression of expressive bodies using moderation. In *International Conference on 3D Vision (3DV)*, pages 792–804, 2021.

[17] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *Transactions on Graphics, (Proc. SIGGRAPH)*, 40(4):1–13, 2021. 2, 3

[18] Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J. Black, and Timo Bolkart. Capturing and animation of body and clothing from monocular video. In *SIGGRAPH Asia 2022 Conference Papers*, New York, NY, USA, 2022. Association for Computing Machinery. 2, 3, 5

[19] Yao Feng, Weiyang Liu, Timo Bolkart, Jinlong Yang, Marc Pollefeys, and Michael J. Black. Learning disentangled avatars with hybrid 3d representations. *arXiv*, 2023. 2, 3

[20] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial NeRF models from monocular video. *Transactions on Graphics (TOG)*, 41(6):1–12, 2022. 3

[21] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schoenborn, and Thomas Vetter. Morphable face models - an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82, Los Alamitos, CA, USA, 2018. IEEE Computer Society. 3

[22] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18653–18664, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2, 3

[23] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *Transactions on Graphics (TOG)*, 38(6): 1–19, 2019. 3

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020. 4

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

[26] Yangyi Huang, Yuliang Xiu, Hongwei Yi, Tingting Liao, Jiaxiang Tang, Deng Cai, and Justus Thies. TeCH: Text-guided Reconstruction of Lifelike Clothed Humans. *arXiv preprint: 2308.08545*, 2023. 3

[27] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 867–876, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 7

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 9

[30] 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. 9

[31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, pages 12888–12900, Baltimore, MD, USA, 2022. PMLR. 2, 6, 9

[32] Junxuan Li, Shunsuke Saito, Tomas Simon, Stephen Lombardi, Hongdong Li, and Jason Saragih. Megane: Morphable eyeglass and avatar network, 2023. 3

[33] 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

[34] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *Transactions on Graphics (TOG)*, 36(6):1–17, 2017. 3

[35] 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, 2023. 2, 3

[36] Weiyang Liu, Zhen Liu, Liam Paull, Adrian Weller, and Bernhard Schölkopf. Structural causal 3d reconstruction. In *ECCV*, 2022. 2

[37] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, Los Alamitos, CA, USA, 2015. IEEE Computer Society. 6

[38] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 3

[39] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *Transactions on Graphics (TOG)*, 34 (6):1–16, 2015. 3

[41] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2, 5

[42] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures, 2022. 2, 3, 6, 7

[43] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13492–13502, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[44] 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. *Commun. ACM*, 65(1):99–106, 2021. 2, 4

[45] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*, New York, NY, USA, 2022. Association for Computing Machinery. 3

[46] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning (ICML)*, pages 16784–16804, Virtual, 2022. PMLR. 3

[47] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10967–10977, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 2, 3, 9

[48] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 2, 3, 4

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, Virtual, 2021. PMLR. 3, 7

[50] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 3

[51] Siddhant Ranade, Christoph Lassner, Kai Li, Christian Haene, Shen-Chi Chen, Jean-Charles Bazin, and Sofien Bouaziz. Ssdnerf: Semantic soft decomposition of neural radiance fields, 2022. 3

[52] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes, 2023. 2, 3, 4, 5, 7

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2, 3, 4, 5, 9

[54] Radu Alexandru Rosu, Shunsuke Saito, Ziyan Wang, Chenglei Wu, Sven Behnke, and Giljoo Nam. Neural strands:

Learning hair geometry and appearance from multi-view images. In *Computer Vision – ECCV 2022*, pages 73–89, Cham, 2022. Springer Nature Switzerland. 3

[55] Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18603–18613, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[56] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 7

[57] Mike Seymour, Chris Evans, and Kim Libreri. Meet mike: Epic avatars. In *ACM SIGGRAPH 2017 VR Village*, New York, NY, USA, 2017. Association for Computing Machinery. 3

[58] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor. *arXiv preprint arXiv:2305.20082*, 2023. 3

[59] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *International Conference on Computer Vision (ICCV)*, pages 3735–3744, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 2, 3

[60] Kevin Turner. Decoding latents to RGB without upscaling, 2022. 6, 9

[61] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3835–3844, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[62] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12619–12629, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 7

[63] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, and Baining Guo. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4563–4573, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 3

[64] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 3

[65] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18072–18081, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 6

[66] Menghua Wu, Hao Zhu, Linjia Huang, Yiyu Zhuang, Yuanxun Lu, and Xun Cao. High-fidelity 3d face generation from natural language descriptions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4521–4530, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 3

[67] Longwen Zhang, Qiwei Qiu, Hongyang Lin, Qixuan Zhang, Cheng Shi, Wei Yang, Ye Shi, Sibei Yang, Lan Xu, and Jingyi Yu. Dreamface: Progressive generation of animatable 3d faces under text guidance, 2023. 2, 3

[68] Rui Zhao, Wei Li, Zhipeng Hu, Lincheng Li, Zhengxia Zou, Zhenwei Shi, and Changjie Fan. Zero-shot text-to-parameter translation for game character auto-creation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21023, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 3

[69] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I m avatar: Implicit morphable head avatars from videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13545–13555, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2, 3

[70] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars, 2022. 2

[71] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. In *Computer Vision – ECCV 2022*, pages 250–269, Cham, 2022. Springer Nature Switzerland. 2, 3