Text and Image Guided 3D Avatar Generation and Manipulation

Zehranaz Canfes* M. Furkan Atasoy* Alara Dirik* Pinar Yanardag
Boğaziçi University
Istanbul, Turkey
{zehranaz.canfes, muhammed.atasoy, alara.dirik}@boun.edu.tr, yanardag.pinar@gmail.com

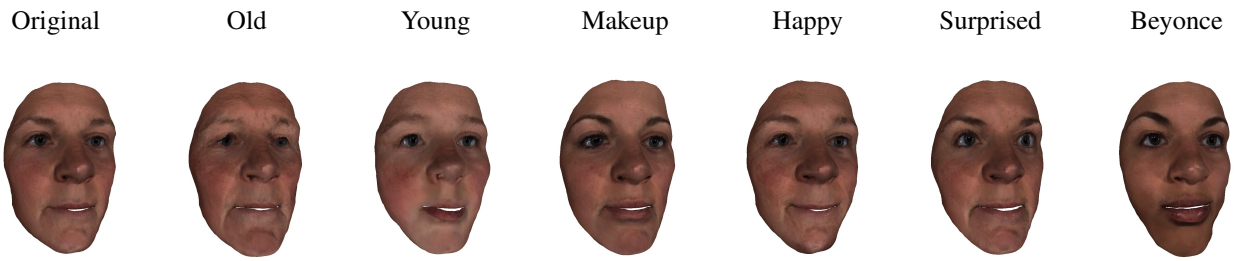


Figure 1. Given an input vector of a 3D mesh (denoted as *Original*) our method modifies the latent code such that the target attribute specified by a text prompt such as *'Young'* or *'Surprised'* is present or enhanced, while leaving other attributes largely unaffected.

Abstract

The manipulation of latent space has recently become an interesting topic in the field of generative models. Recent research shows that latent directions can be used to manipulate images towards certain attributes. However, controlling the generation process of 3D generative models remains a challenge. In this work, we propose a novel 3D manipulation method that can manipulate both the shape and texture of the model using text or image-based prompts such as 'a young face' or 'a surprised face'. We leverage the power of Contrastive Language-Image Pre-training (CLIP) model and a pre-trained 3D GAN model designed to generate face avatars, and create a fully differentiable rendering pipeline to manipulate meshes. More specifically, our method takes an input latent code and modifies it such that the target attribute specified by a text or image prompt is present or enhanced, while leaving other attributes largely unaffected. Our method requires only 5 minutes per manipulation, and we demonstrate the effectiveness of our approach with extensive results and comparisons. Our project page and source code can be found at https://catlab-team.github.io/latent3D.

*Equal contribution.

1. Introduction

Generative Adversarial Networks (GAN) for 2D vision have achieved several breakthroughs, such as Progressive GAN [30], BigGAN [5], and StyleGAN [32, 33], which enable high-resolution and high-quality image generation in various domains. 3D vision and the field of 3D generation have made similarly remarkable progress, with the development of implicit surface and volume representations [7, 37, 39] enabling the encoding, reconstruction, and generation of detailed models of watertight surfaces without suffering from the limitations of using a 3D grid or fixed-topology meshes. While these implicit representation-based approaches result in a learnable surface parameterization that is not limited in resolution, they often require coordinate sampling for non-differentiable point cloud and mesh generation, which is also time consuming. Other works such as UV-GAN [9], GANFit [15], and TBGAN [14] restrict the 3D generation problem to a 2D domain and aim to generate 3D shapes by training GANs directly on UV maps of shapes, normals, and textures. Despite advances in 3D generation, the question of how to control the re-

sults of 3D generative models remains an active research topic. The issue of 3D manipulation is particularly important in morphable models such as the human face and body, where a natural consequence of this work is to enable animation. Previous work, known as 3D morphable models (3DMM) [3, 6], represent 3D faces as disentangled PCA models of geometry, expression, and texture, and manipulates faces by editing each modality separately. However, the linear nature of PCA makes it difficult to generate novel faces and high-quality reconstructions. Moreover, numerous previous work use 3DMMs as backbone models and attempt to reconstruct 3D faces from 2D images or partial face scans [26–28, 58] and thus suffer from their fundamental limitations. In contrast, there have been significant advances in the manipulation of rigid 3D objects in recent years. Several methods have been proposed for manipulating implicit 3D shape representations with text [38, 46, 56] and sketches [18]. However, these methods require hours of optimization per manipulation and are limited to rigid and simple shapes such as *chairs*, while we attempt to manipulate articulated, diverse, and complex shapes such as the *human face*.

In this work, we propose a method for fast and highly effective text and image-driven 3D manipulation of facial avatars. Our method uses a pre-trained generative 3D model, TBGAN, as a base GAN model and leverages the joint image-text representation capabilities of Contrastive Language-Image Pre-training (CLIP) [42] to optimize a latent code based on a user-provided text or image prompt (see Figure 1 for example manipulations). Unlike previous work [38, 46, 56], which require a large amount of time, our method requires only 5 minutes per manipulation and enables text- and image-driven edits that allow precise, fine-grained, and complex manipulations such as modifying the *gender* and *age* attributes without affecting the irrelevant attributes or identity of the original mesh. Our proposed method directly optimizes the shape, normal, and texture images and performs disentangled manipulations. Furthermore, we propose a baseline method that uses PCA to detect unsupervised latent directions on facial avatars. Our experiments show that our method is able to outperform PCA-based baseline and TBGAN on various simple and complex manipulations.

2. Related Work

2.1. 3D Shape Representations

Unlike 2D vision problems, where RGB images have become almost the standard data format, how to best represent 3D data remains an active research question. As a result, a variety of representations are used in work on 3D vision problems, such as point clouds, voxels, meshes, and more recently, neural implicit representations.

One of the most popular 3D data formats is point clouds, which are lightweight 3D representations consisting of coordinate values in (x, y, z) format. They are widely used in 3D learning problems such as 3D shape reconstruction [13, 36, 41, 50], 3D object classification [13, 41], and segmentation [41]. However, point clouds provide limited information about how points are connected and pose view-dependency issues. Another 3D format, triangular mesh, describes each shape as a set of triangular faces and connected vertices. Meshes, while better suited to describing the topology of objects, often require advanced preprocessing steps to ensure that all input data has the same number of vertices and faces. The voxel format describes objects as a volume occupancy matrix, where the size of the matrix is fixed. While the voxel format is well suited for CNN-based approaches, it requires high resolution to describe fine-grained details. Finally, numerous neural implicit representations have been proposed in recent years to overcome the shortcomings of classical representations. These methods represent 3D shapes as deep networks that map 3D coordinates to a signed distance function (SDF) [39] or occupancy fields [7, 37] to create a lightweight, continuous shape representation. However, a major drawback of implicit representations is that they require aggressive sampling and querying of 3D coordinates to construct surfaces. Finally, works such as UV-GAN [9], GANFit [15], and TBGAN [14] represent shapes and textures as 2D positional maps that can be projected back into 3D space, and leverage recent advances in 2D imaging [30] to jointly generate novel face shapes and textures. In our work, we use TBGAN as our base generative model due to its speed and generative capabilities.

2.2. Latent Space Manipulation

Latent space manipulation methods for image editing can be categorized into supervised and unsupervised methods. Supervised approaches typically leverage pre-trained attribute classifiers or train new classifiers using labeled data to optimize a latent vector and enhance the target attribute's presence in the generated image [16, 48]. On the other hand, several unsupervised approaches have been proposed to show that it is possible to find meaningful directions in the latent space of large-scale GANs without using classifiers or labeled data [24, 55]. For instance, GANSpace [22] proposes applying principal component analysis (PCA) [57] on a set of randomly sampled latent vectors extracted from the intermediate layers of BigGAN and StyleGAN. SeFA [49] proposes a similar approach that directly optimizes the intermediate weight matrix of the GAN model in closed form. A more recent work, LatentCLR [60], proposes a contrastive learning approach to find unsupervised directions that are transferable to different classes. Moreover, StyleCLIP [40] and StyleMC [34] both propose using

CLIP for text-guided manipulation of both randomly generated and encoded images with StyleGAN2. These methods show that it is possible to use CLIP for fine-grained and disentangled manipulations of images.

2.3. 3D Shape Generation and Manipulation

In recent years, there have been tremendous advances in 3D shape generation. While some of this work includes traditional 3D representations such as point clouds [1, 13, 23, 41, 51], voxels [8, 29], and meshes [17, 20, 21], several approaches have been proposed to use implicit surface and volume representations for high-quality and scalable representations (see [7, 37, 39]). However, most of this work focuses on generating rigid objects, a relatively simple task compared to the generation of articulate, morphable shapes such as the human face and body. In contrast to rigid object generation, most work on human face generation and reconstruction uses linear statistical models known as 3DMMs. [3] which trains separate linear statistical models using PCA for face shape, expression, and texture on a dataset of registered face meshes where the corresponding keypoints are available. However, the linear nature of PCA makes it difficult to perform high-quality reconstructions and novel manipulations. Several works on face generation address this problem and propose various methods with limited success (see [4, 11, 53, 54]). In addition, 3DMM is widely used as the backbone of various applications such as 3D reconstruction of faces from multiple images [2, 43, 45, 52] or a single image [25, 47, 59].

Despite the advances in 3D generation, the work on 3D shape manipulation is much more limited and focuses on supervised or unsupervised manipulation of shapes or textures separately. Unsupervised 3D manipulation methods often introduce additional constraints into the training process to enforce better controllability [12, 35]. In addition, several supervised 3D manipulation methods have recently been proposed. Text2Mesh [38] proposes a neural style model that encodes the style of a single mesh and uses a CLIP -based method for texture manipulations. However, this method requires training a separate model for each shape to be manipulated and is limited to texture manipulations. Another work, [56], proposes a CLIP -based method for text- and image-based manipulation of NeRFs. However, this method requires training multiple models per text to map the CLIP embedding of an input image or text onto the latent space of the proposed deformation network. Similarly, CLIP -Forge [46] trains an auto-encoding occupancy network and a normalizing flow model to connect the CLIP encodings of 2D renders of simple 3D shapes such as *chair* or *table* and the latent shape encodings. We note that this method is limited to shape manipulation of simple rigid objects and does not allow for high-resolution generation or fine-grained edits due to the use of the voxel format. In addition, this method requires more than 2 hours per manipulation. Unlike other CLIP -based 3D manipulation methods, our method can manipulate both shape and texture and requires only 5 minutes to perform complex and accurate manipulations of articulated face avatars.

3. Methodology

In this section, we first briefly describe TBGAN and then introduce our method for text and image-driven manipulation of 3D objects.

3.1. Background on TBGAN

In our work, we use TBGAN as a generative base model for manipulation, a GAN model with an architecture that closely resembles PG-GAN [31]. More specifically, TBGAN proposes a progressively growing architecture that takes a one-hot- encoded facial expression vector $\mathbf{e}$ encoding 7 universal expressions *neutral, happy, angry, sad, afraid, disgusted, and surprised* and a random noise vector $\mathbf{z}$ as input. It then progressively generates higher-dimensional intermediate layer vectors known as *modality correlation* layers, and branches to *modality-specific* layers to jointly generate high-quality shape, shape-normal, and texture images. The model is trained on large-scale, high-resolution UV maps of preprocessed meshes with WGAN-GP loss [19]. Within the modality correlation layers, the so-called *trunk network* preserves the modality correspondences, while the separate branches of the modality-specific layers allow learning independent distributions for shape and texture data. Unlike 3DMMs, which are often constructed by applying PCA to datasets of 3D scans from hundreds or thousands of subjects [3], TBGAN is not bound by linear separability constraints and provides a continuous latent space.

3.2. Text and Image Guided Manipulation

Given a pre-trained TBGAN generator $\mathcal{G}$, let $\mathbf{z} \in \mathcal{R}^d$ denote a d-dimensional random input vector sampled from a Gaussian distribution $\mathcal{N}\left(0, \sigma^2\right)$ and $\mathbf{e}$ denote a one-hot-encoded facial expression vector initialized to zero to obtain a neutral expression. Let $\mathbf{c} \in \mathcal{C}$ denote an intermediate layer vector obtained by partial forward propagation of $\mathbf{z}$ and $\mathbf{e}$ through the generator $\mathcal{G}$. Our method first generates a textured mesh by using the generated shape, normal and texture UV maps via cylindrical projection. Then given a text prompt t such as '*happy human*', $\mathbf{c}$ is optimized via gradient descent to find a direction $\Delta\mathbf{c}$, where $\mathcal{G}(\mathbf{c} + \Delta\mathbf{c})$ produces a manipulated textured mesh in which the target attribute specified by t is present or enhanced, while other attributes remain largely unaffected. In our work, we optimize the original intermediate latent vector $\mathbf{c}$ using gradient descent and work in the 4×4/dense layer of the TBGAN generator (see ablation study in Section A.1 on the choice

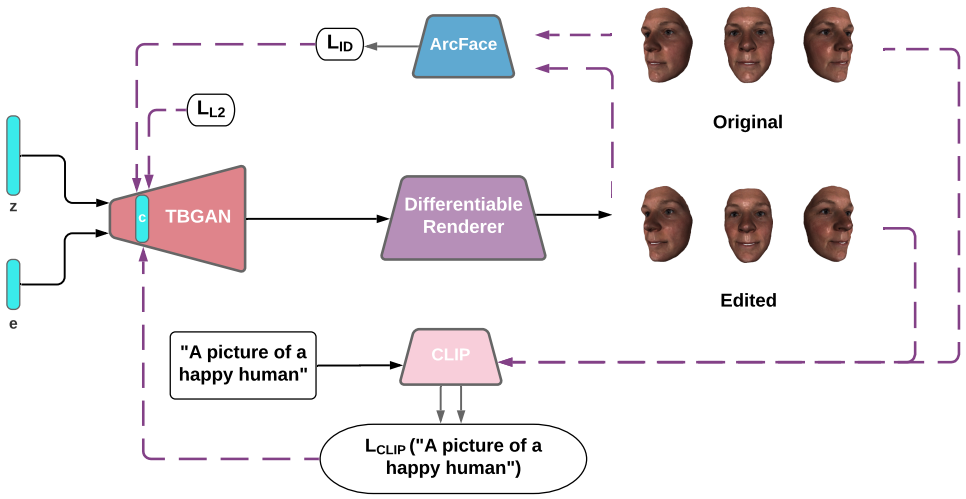


Figure 2. An overview of our framework (using the text prompt *'happy human'* as an example). The manipulation direction $\Delta\mathbf{c}$ corresponding to the text prompt is optimized by minimizing the CLIP-based loss $\mathcal{L}_{\text{CLIP}}$, the identity loss $\mathcal{L}_{\text{ID}}$, and the L2 loss $\mathcal{L}_{\text{L2}}$.

of layer used for manipulation). The optimized latent vector $\mathbf{c} + \Delta\mathbf{c}$ can then be fed into TBGAN to generate shape, normal, and texture UV maps, and finally a manipulated mesh with the target attributes. A diagram of our method is shown in Figure 2. To perform meaningful manipulation of meshes without creating artifacts or changing irrelevant attributes, we use a combination of a CLIP-based loss, an identity loss, and an L2 loss as follows:

$$\arg\min_{\Delta\mathbf{c}\in\mathcal{C}} \mathcal{L}_{\text{CLIP}} + \lambda_{\text{ID}}\mathcal{L}_{\text{ID}} + \lambda_{\text{L2}}\mathcal{L}_{\text{L2}} \tag{1}$$

where λ_{ID} and λ_{L2} are the hyperparameters of $\mathcal{L}_{\text{ID}}$ and $\mathcal{L}_{\text{L2}}$, respectively. While CLIP-based loss ensures that the user-specified attribute is present or enhanced, ID-loss and L2-loss leave other attributes unchanged, forcing disentangled changes. The identity loss $\mathcal{L}_{\text{ID}}$ minimizes the distance between the identity of the original renders and the manipulated renders:

$$\mathcal{L}_{\text{ID}} = 1 - \langle R(\mathcal{G}(\mathbf{c})), R(\mathcal{G}(\mathbf{c} + \Delta\mathbf{c}))\rangle \tag{2}$$

where R is ArcFace [10], a facial recognition network in the case of face recognition, and $\langle\cdot,\cdot\rangle$ computes the cosine similarity between the identities of the rendered image and the manipulated result (see ablation study in Section A.2 on the effect of identity loss). The L2 loss is used to prevent artifact generation and defined as:

$$\mathcal{L}_{\text{L2}} = ||\mathbf{c} - (\mathbf{c} + \Delta\mathbf{c})||_2 \tag{3}$$

The CLIP -based loss term $\mathcal{L}_{\text{CLIP}}$ can be defined in two different ways, depending on the type of prompt provided by the user: the user can either provide text prompts such as *'old human'* or a target image such as an image of *Bill Clinton* for manipulation. If the user provides a list of text prompts, the CLIP-based loss is given by:

$$\mathcal{L}_{\text{CLIP}} = \frac{\sum_{j=1}^{K}\sum_{i=1}^{N} D_{\text{CLIP}}(\mathcal{I}_i, t_j)}{K \cdot N} \tag{4}$$

where $\mathcal{I}_i$ is a rendered image from a list of N rendered images, t_j is the target text t embedded in a text template from a list of K templates. Here, $\mathcal{L}_{CLIP}$ is used to minimize the cosine distance between CLIP embeddings of the rendered images I_i and the set of text prompts t_j. In the case where the user specifies a target image, the CLIP-based loss is given by:

$$\mathcal{L}_{\text{CLIP}} = \frac{\sum_{i=1}^{N} D_{\text{CLIP}}(\mathcal{I}_i, \mathcal{I}_{\text{targ}})}{N} \tag{5}$$

where $\mathcal{I}_{\text{targ}}$ is the target image. Here, $\mathcal{L}_{\text{CLIP}}$ seeks to minimize the cosine distance between CLIP embeddings of the rendered images $\mathcal{I}_i$ and the target image $\mathcal{I}_{\text{targ}}$. We use D_{CLIP} to compute the cosine distance between CLIP embeddings in both methods. The renderings $\mathcal{I}_i$ and templates t_j are created as follows.

Differentiable rendering Note that to optimize the latent code $\mathbf{c}$, we need to compute the CLIP distance between the given text prompt and the generated mesh. Since the CLIP model cannot handle 3D meshes, we render the generated mesh corresponding to the latent code $\mathbf{c}$ in 2D and feed it with the pre-trained CLIP model. However, this simple strategy is not sufficient to optimize the latent code $\mathbf{c}$ via gradient descent, since the rendering operation needs to be differentiable. To this end, we create a fully differentiable pipeline using an open-source differentiable renderer

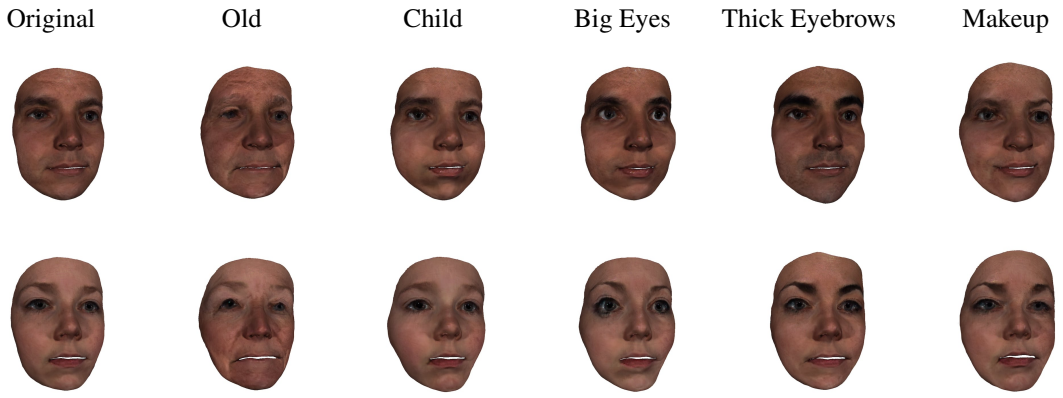


Figure 3. Manipulation results of our method with various inputs and text prompts on two different 3D faces: 'Old', 'Child', 'Big Eyes', 'Thick Eyebrows', 'Makeup'. The leftmost column shows the original outputs, the adjacent columns show the manipulation results, the target text prompt is above each column.

library, PyTorch3D [44]. To enforce view consistency, we render N=3 views of the generated mesh from an anchor, where the mesh $\mathcal{M}$ is rotated by -30, 3, and 30 degrees: $f_{\text{diff}}(\mathcal{M}, \theta_{cam}) = \mathcal{I}$, where θ_{cam} denotes the camera, object position, and rotation parameters used by the differentiable renderer. We denote the renders generated by f_{diff} as $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$. We compute the CLIP-based loss for each image and average them prior to feeding them into our loss, which leads to more stable results.

Prompt engineering Our model takes a user-defined text prompt t as input which describes the target manipulation. Previous work has shown that using prompt engineering [42] to augment the text prompts yield more consistent results. Hence, we augment the original text prompt t by embedding it in sentence templates such as '*a rendering of a* ...' or '*a face of a* ...' to generate K text prompts $t_1, t_2, ..., t_k$ (see Appendix B for the full list of templates used in this work). Note that more than one semantically equivalent text prompt can be given as input to achieve more stable results. For example, to achieve a manipulation that makes the face look *older*, our method can use a list of different text prompts such as '*old human*' and '*aged person*'.

4. Experiments

In this section, we present the experimental results of our proposed method and evaluate our results based on manipulation quality and fidelity. In addition, we present a simple baseline method for manipulation of 3D objects using PCA, and show that our method allows for more disentangled manipulations without changing the identity of the manipulated face. Furthermore, we compare the manipulation performance of our method and TBGAN on a list of face expressions.

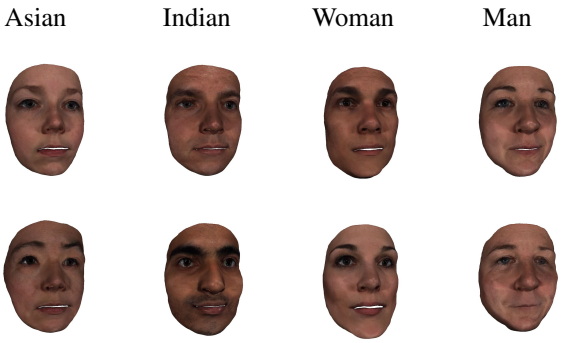


Figure 4. Manipulation results of our method with various inputs and text prompts: 'Asian', 'Indian', 'Woman', 'Man'. The top row shows the original outputs, the bottom row shows the manipulation results, target text prompt is above each column.

4.1. Experimental Setup

We use the official implementation and pre-trained model of TBGAN [1]. For all manipulation experiments, we first generate random meshes using TBGAN and its default hyperparameters. For differentiable rendering, we use the renderer of PyTorch3D without additional training or fine-tuning and render 3 images at each generation step with mesh y-axis angles set to -30, 3, and 30, respectively. The rest of the renderer hyperparameters are as follows: We set the blur radius to 0.0, faces per pixel to 2.0, and the position of the point lights is set to $(0.0,\ 0.0,\ +3.0)$. For all manipulation experiments with CLIP, we set the loss terms as follows: $\lambda_{\text{ID}} = 0.01$, $\lambda_{\text{L2}} = 0.001$. We use a fixed number of 100 optimization steps for each manipulation. We also use the Adam optimizer and keep the default hyperparameters. We run all experiments on a TITAN RTX GPU.

[1] https://github.com/barisgecer/TBGAN

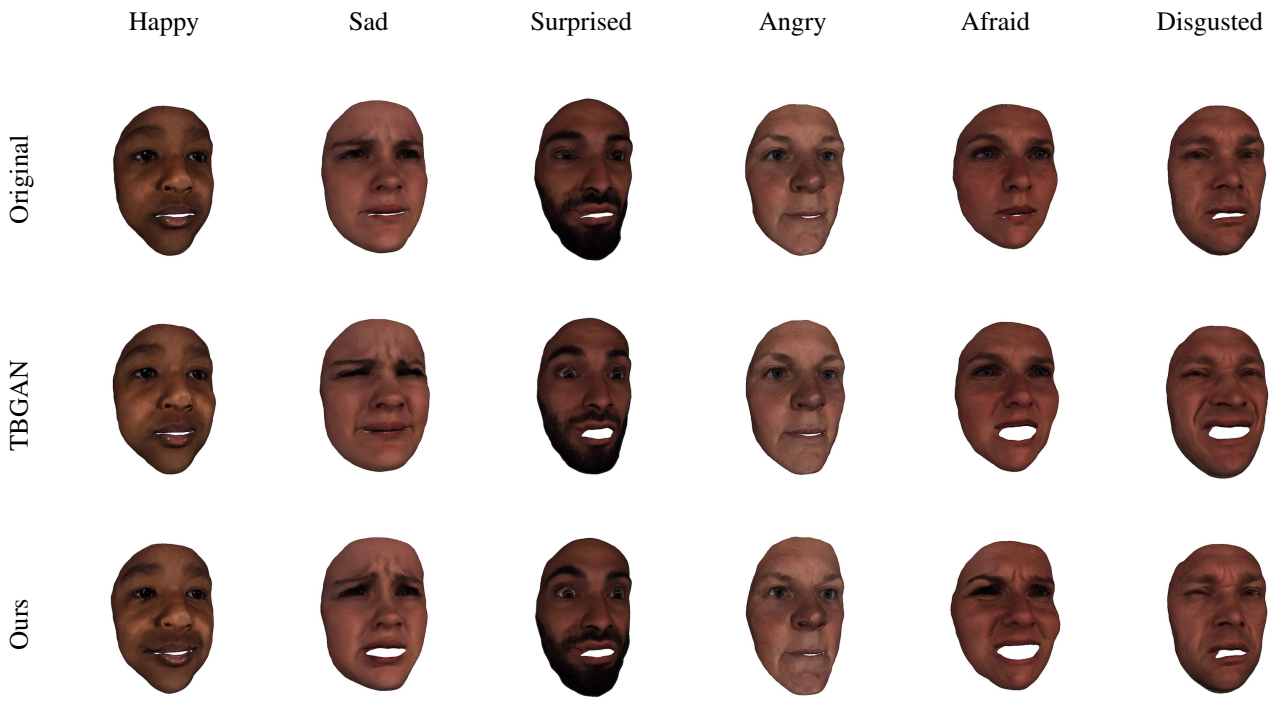


Figure 5. The results of our method with text prompts: 'Happy', 'Sad', 'Surprised', 'Angry', 'Afraid', and 'Disgusted'. The top row shows the original outputs, the second row shows the TBGAN conditioned expressions, the third row shows the manipulation results, the target text prompt is above each column.

4.2. Qualitative Results

In this section, we demonstrate the quality and consistency of the results obtained by our method on a diverse set of generated faces. We start with simple manipulations such as *'big eyes'* and continue with complex text prompts such as *'Asian'*. We then continue with manipulations on facial expressions and then share the qualitative results for the image-based manipulations.

Results on Simple and Complex Text Prompts We begin with a series of text-based manipulations, ranging from simple attributes such as *'big eyes'*, *'thick eyebrows'*, *'makeup'* to fine-grained attributes such as *'old'*, *'child'*, and present the results in Figure 3. As can be seen in the figure, our method successfully performs the targeted manipulations on various faces and produces details with high granularity while preserving the global semantics and underlying content. For example, the manipulated outputs for the target text prompt *'old'* represent elderly people, while preserving the overall shape and identity of the original meshes. We also show that our method provides global semantic understanding of more complex attributes such as *'man'*, *'woman'*, *'Asian'*, and *'Indian'*. Figure 4 shows the results for manipulations on various randomly generated outputs, where we can see that our method is able to perform complex edits such as *ethnicity* and *gender*.

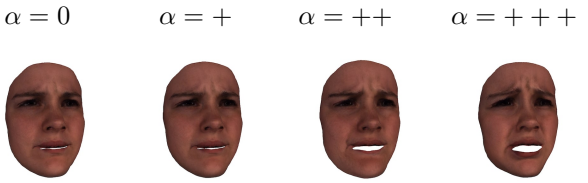


Figure 6. The results for different manipulation strengths for the text prompt *'sad'*. $\alpha = 0$ represents the original image, while $\alpha+$ to $\alpha + ++$ represent increasing manipulation strength.

Results on Facial Expressions The manipulation capabilities of our method are not limited to the physical characteristics of the generated avatars, but can also be used to change their facial expressions such as *'smiling'*, *'angry'*, and *'surprised'*. As can be seen in Figure 5, our method

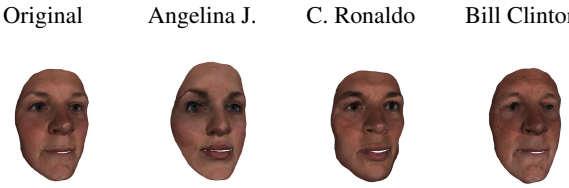


Figure 7. Results for text-prompts 'Angelina Jolie', 'Cristiano Ronaldo' and 'Bill Clinton' using our method.

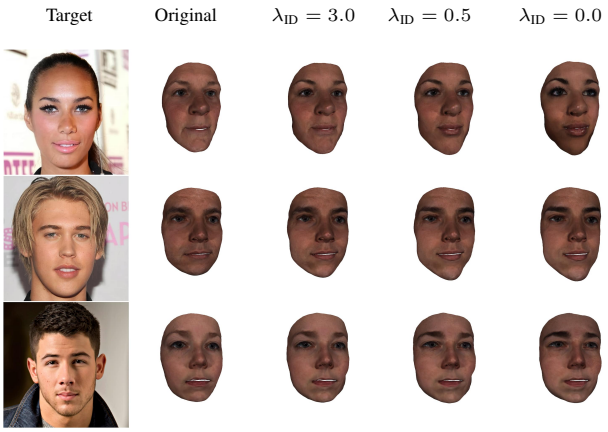


Figure 8. Image-based manipulations with the target image on the left and manipulations with different strengths for identity loss on the right.

can successfully manipulate a variety of complex emotions on various input meshes with almost no change to other attributes. Moreover, our model is able to control the intensity of the expression by increasing α such that $\mathbf{c} + \alpha\Delta\mathbf{c}$. As α is increased, the extent of expression changes, as you can see in Figure 6.

Results on Identity Manipulations In addition to general manipulations of physical attributes and expressions, we demonstrate that our method can be used to perform complex identity manipulations with solely text-based manipulations. For this experiment, we use text manipulations with the prompts 'Bill Clinton', 'Cristiano Ronaldo', and 'Angelina Jolie' and show the results in Figure 7. As can be seen in the figure, our method is able to achieve a manipulation that captures the characteristics of the target person, such as Bill Clinton's characteristic droopy eyes or Ronaldo's jaw structure.

Results on Image-Guided Manipulations We note that our method is not limited to text-based manipulation, but can also be used to manipulate a mesh with a target image, as described in Section 3.2. We note that unlike text-based manipulations, image-based manipulations inevitably alter

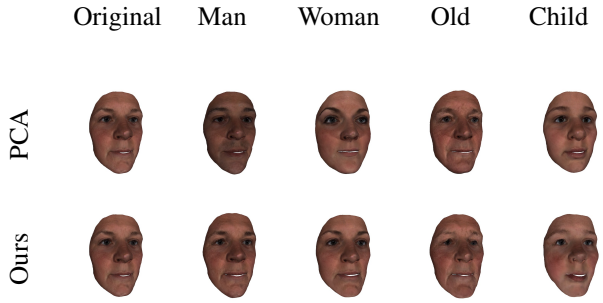


Figure 9. Comparison between PCA-based manipulations and text-driven manipulations using our method. The top row shows the PCA-based results, the bottom row shows the results with our method.

both coarse and fine-grained attributes. Figure 8 shows the manipulation results for different images of celebrities with $\lambda_{\text{L2}} = 0$ and $\lambda_{\text{ID}} = \{3.0, 0.5, 0\}$. As can be seen in the figure, our method can capture complex identity-related attributes of the target image regardless of illumination and pose, and perform successful manipulations such as capturing *skin color, eye makeup, eyebrow* and *jaw structure*. We can also observe that the manipulations with $\lambda_{\text{ID}} = 3.0$ produce a face that resembles both the original image and the target, while $\lambda_{\text{ID}} = 0.0$ reconstructs the target image in 3D.

4.3. Comparison with PCA

Our method performs a text-driven manipulation of facial meshes. Since there are no existing approaches for this task, we propose a simple PCA-based baseline inspired by GANSpace [22]. For this experiment, we sample $10,000$ intermediate latent vectors from the 4×4/dense layer of TBGAN and apply PCA to the concatenated latent vectors to obtain principal components, where each component represents a new transformed axis that is a linear combination of the original features. Using the found directions, we can manipulate the original latent vectors by directly applying a principal component, i.e., using the component as a direction vector to steer the generation:

$$c^{'} = c + (\alpha \times n \times \text{PC}_i) \tag{6}$$

Here α denotes the step size, n the number of steps, and PC_i the i_{th} principal component used. For comparison purposes, we keep the top ranked principal components and apply to them randomly generated latent vectors with a step size of α of 10. We note that the top-ranking principal components encode prominent directions such as *age* and *gender*. For comparison, we apply age and gender-based text-driven edits to the same latent vectors and present the comparative results in Figure 9. As can be seen in the figure, the top directions encoded by PCA, *age* and *gender*,

Accuracy	Male	Female	Old	Young	All
PCA	**4.42** ±0.65	**4.13** ±0.99	4.25 ±0.61	4.21 ±0.83	4.25 ±0.78
Ours	4.21 ±0.83	4.08 ±0.97	**4.67** ±0.56	**4.46** ±0.78	**4.35** ±0.82
Identity	Male	Female	Old	Young	All
PCA	2.85 ±1.04	2.57 ±1.08	3.52 ±1.29	3.01 ±1.30	2.99 ±1.21
Ours	**4.45** ±0.76	**4.33** ±0.80	**3.70** ±1.22	**3.62** ±1.20	**4.02** ±1.07

Table 1. Mean scores (1-5) for identity preservation and manipulation accuracy for our method and PCA-based baseline.

	TBGAN	Ours
Surprise	4.64 ±0.48	**4.70** ±0.46
Disgust	4.04 ±1.12	**4.26** ±0, 61
Happy	1.83 ±0.92	**3.77** ±0.90
Sad	4.17 ±0.87	**4.48** ±0.71
Afraid	2.87 ±1.08	**3.43** ±0.97
Angry	2.05 ±1.22	**3.65** ±1.24
All	3.26 ±1.46	**4.05** ±0.97

Table 2. Mean scores (1-5) and std values for manipulation accuracy on various expressions using our method and TBGAN.

significantly alter the identity of the input person, while our method achieves the desired manipulations without altering irrelevant attributes.

Human Evaluation We also conduct human evaluations to measure the perceived quality of the generated results. More specifically, we are interested in the extent to which the manipulation results match the input target text and preserve the other attributes. For the human evaluation, we asked $n = 25$ users to evaluate 5 randomly selected sets of input meshes, text prompts t, and manipulated meshes. For each set, we display the target text and output in pairs and ask users to assign a score between $[1, 5]$ for two questions: *'How well does the manipulation achieve the attributes specified in the target text?'* and *'Is the identity of the input face preserved?'*. In Table 1, we report the mean and standard deviations of the scores for the questions (denoted as *Accuracy* and *Identity*, respectively). As the human-evaluation results show, our method performs better than PCA in all settings except the Female and Male attributes. However, we note that our method performs significantly better than PCA on these attributes when it comes to identity preservation, which was evaluated by human raters. More specifically, the human raters found that our method preserves identity while achieving competitive attribute preservation scores that are 34% higher than PCA.

4.4. Comparison with TBGAN

TBGAN does not provide a way to manipulate the generated meshes, but it is possible to obtain different facial expressions by modifying one-hot encoded expression vector of TBGAN. Therefore, we compare the manipulations done by our method on the facial expressions with TBGAN. As can be seen in Figure 5, our results are more successful in terms of realistic facial expression representation. We also performed a human evaluation to compare the performance of facial expression manipulations by asking $n = 25$ participants; *"How well does the manipulation achieve the expressions specified in the target text?"*. Table 2 shows the mean scores and standard deviations of the results, with our method outperforming TBGAN in all settings. Moreover, our method is able to gradually change the expression (see Figure 6) which is not possible using TBGAN since it only produces a fixed expression using the one-hot encoded vector.

5. Limitations and Broader Impact

While our method is very effective for both coarse and fine-grained manipulations, our base generative model is trained to generate partial face avatars. We therefore strongly believe that our work can be extended to more comprehensive generation scenarios to produce full head meshes or bodies.

6. Conclusion

We proposed an approach to manipulating 3D facial avatars with text and image inputs that relies on a differentiable rendering pipeline combined with CLIP -based and identity-based losses. Unlike previous work that limits the manipulation of 3D shapes to either local geometric changes such as texture or only shape, our method can perform high-level and complex manipulations of shapes and textures. Another major advantage of our method is that it requires only 5 minutes to manipulate a given mesh, while other works require an hour of optimization time per manipulation. Given that avatar and human body generation is widely used in industries such as character design, animation, and visual effects, we see two natural improvements for our work: sketch-based manipulation for more intuitive and user-friendly manipulations, and the extension of our framework to full-body generation.

Acknowledgments This research is produced with support from the 2232 International Fellowship for Outstanding Researchers Program of TUBITAK (Project No: 118c321).

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, 2018. 3

[2] Ziqian Bai, Zhaopeng Cui, Jamal Ahmed Rahim, Xiaoming Liu, and Ping Tan. Deep facial non-rigid multi-view stereo. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5849–5859, 2020. 3

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99*, 1999. 2, 3

[4] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models "in-the-wild". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5464–5473, 2017. 3

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018. 1

[6] Alan Brunton, Augusto Salazar, Timo Bolkart, and Stefanie Wuhrer. Review of statistical shape spaces for 3d data with comparative analysis for human faces. *Comput. Vis. Image Underst.*, 128:1–17, 2014. 2

[7] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5932–5941, 2019. 1, 2, 3

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 3

[9] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7093–7102, 2018. 1, 2

[10] Jiankang Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019. 4

[11] Chi Nhan Duong, Khoa Luu, Kha Gia Quach, and Tien D. Bui. Beyond principal components: Deep boltzmann machines for face modeling. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4786–4794, 2015. 3

[12] Tim Elsner, Moritz Ibing, Victor Czech, Julius Nehring-Wirxel, and Leif P. Kobbelt. Intuitive shape editing in latent space. *ArXiv*, abs/2111.12488, 2021. 3

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2, 3

[14] Baris Gecer, Alexandros Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *European Conference on Computer Vision*, pages 415–433. Springer, 2020. 1, 2

[15] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1164, 2019. 1, 2

[16] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5744–5753, 2019. 2

[17] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[18] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and P. Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. *ArXiv*, abs/2104.00482, 2021. 2

[19] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 3

[20] Zhizhong Han, Zhenbao Liu, Junwei Han, Chi-Man Vong, Shuhui Bu, and Chun Lung Philip Chen. Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3-d meshes. *IEEE transactions on neural networks and learning systems*, 28(10):2268–2281, 2016. 3

[21] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38:1 – 12, 2019. 3

[22] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 2, 7

[23] Le Hui, Rui Xu, Jin Xie, Jianjun Qian, and Jian Yang. Progressive point cloud deconvolution generation network. In *ECCV*, 2020. 3

[24] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019. 2

[25] Luo Jiang, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu. 3d face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing*, 27:4756–4770, 2018. 3

[26] Amin Jourabloo and Xiaoming Liu. Pose-invariant 3d face alignment. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3694–3702, 2015. 2

[27] Amin Jourabloo and Xiaoming Liu. Large-pose face alignment via cnn-based dense 3d model fitting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4188–4196, 2016. 2

[28] Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Pose-invariant face alignment with a single cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3219–3228, 2017. 2

[29] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1966–1974, 2015. 3

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. 1, 2

[31] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2018. 3

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 1

[33] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020. 1

[34] Umut Kocasari, Alara Dirik, Mert Tiftikci, and Pinar Yanardag. Stylemc: Multi-channel based fast text-guided image generation and manipulation. *ArXiv*, abs/2112.08493, 2021. 2

[35] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ArXiv*, abs/2108.04476, 2021. 3

[36] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*, 2018. 2

[37] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2019. 1, 2, 3

[38] Oscar Jarillo Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *ArXiv*, abs/2112.03221, 2021. 2, 3

[39] Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 1, 2, 3

[40] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021. 2

[41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2, 5

[43] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia Giraldez, Xavier Giró i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. *ArXiv*, abs/2107.12512, 2021. 3

[44] Nikhila Ravi, Jeremy Reizenstein, David Novotný, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *SIGGRAPH Asia 2020 Courses*, 2020. 5

[45] Joseph Roth, Y. Tong, and Xiaoming Liu. Adaptive 3d face reconstruction from unconstrained photo collections. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4197–4206, 2016. 3

[46] Aditya Sanghi, Hang Chu, J. Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *ArXiv*, abs/2110.02624, 2021. 2, 3

[47] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1585–1594, 2017. 3

[48] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[49] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 2

[50] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 2

[51] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3858–3867, 2019. 3

[52] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed A. Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Fml: Face model learning from videos. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10804–10814, 2019. 3

[53] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2549–2559, 2018. 3

[54] A. Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1502, 2017. 3

[55] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *In-*

ternational Conference on Machine Learning, pages 9786–9796. PMLR, 2020. 2

[56] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *ArXiv*, abs/2112.05139, 2021. 2, 3

[57] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. 2

[58] Hao Wu, Xiaoming Liu, and Gianfranco Doretto. Face alignment via boosted ranking model. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[59] Yifan Xing, Rahul Tewari, and Paulo R. S. Mendonça. A self-supervised bootstrap method for single-image 3d face reconstruction. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1014–1023, 2019. 3

[60] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14263–14272, October 2021. 2

A. Ablation Study

In this section, we perform ablation studies on the effects of identity loss and layer selection for latent space manipulation.

A.1. Effect of Layer Selection

We perform our manipulations on the 4×4/Dense layer of TBGAN, the layer that provides the best results in terms of identity preservation and meaningful manipulations.

The comparison of our method on different layers can be found in Figure 10. We show that the manipulations on other layers give defected results with undesirable artifacts, so that the results deviate from the desired text prompt.

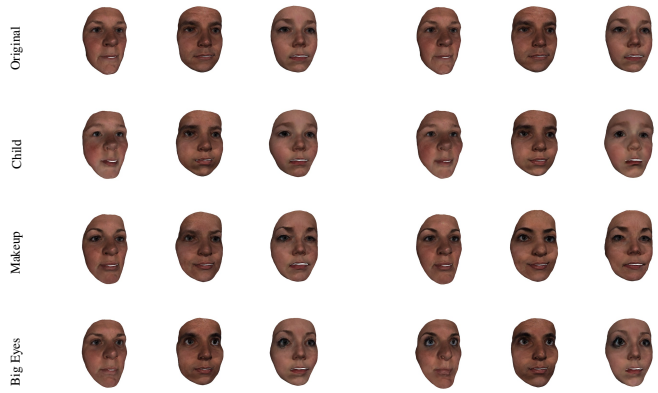
A.2. Effect of Identity Loss

Our method uses ArcFace, a large-scale pre-trained face recognition network, to compute identity loss $\mathcal{L}_{\text{ID}}$ and enforce identity preservation during manipulation. We perform an ablation study with different target texts describing emotion-, shape-, and texture-related changes to demonstrate the effect of $\mathcal{L}_{\text{ID}}$ on the manipulation results, and present the results in Figure 11. For the identity loss experiments, we simply set $\mathcal{L}_{\text{ID}} = 0$ and leave the other hyperparameters the same. As can be seen in Figure 11, identity loss is crucial for preserving the identity of the input, and omitting it leads to manipulation results that are significantly different from the input.

Figure 10. The comparison of manipulations on different layers for two different 3D faces. First two column show the 'beard' and 'old' manipulations on one 3D face and the second column show the results for the same manipulations on another 3D face.

B. Sentence Templates for Prompt Engineering

Our method uses 74 sentence templates. The list of templates we use for augmentation can be found in Table 3.



(a) With ID loss. (b) Without ID loss.

Figure 11. Results with and without ID loss.

‘a bad photo of a’	‘a sculpture of a’
‘a photo of the hard to see’	‘a low resolution photo of the’
‘a rendering of a’	‘graffiti of a’
‘a bad photo of the’	‘a cropped photo of the’
‘a photo of a hard to see’	‘a bright photo of a’
‘a photo of a clean’	‘a photo of a dirty’
‘a dark photo of the’	‘a drawing of a’
‘a photo of my’	‘the plastic’
‘a photo of the cool’	‘a close-up photo of a’
‘a painting of the’	‘a painting of a’
‘a pixelated photo of the’	‘a sculpture of the’
‘a bright photo of the’	‘a cropped photo of a’
‘a plastic’	‘a photo of the dirty’
‘a blurry photo of the’	‘a photo of the’
‘a good photo of the’	‘a rendering of the’
‘a in a video game.’	‘a photo of one’
‘a doodle of a’	‘a close-up photo of the’
‘a photo of a’	‘the in a video game.’
‘a sketch of a’	‘a face of the’
‘a doodle of the’	‘a low resolution photo of a’
‘the toy’	‘a rendition of the’
‘a photo of the clean’	‘a photo of a large’
‘a rendition of a’	‘a photo of a nice’
‘a photo of a weird’	‘a blurry photo of a’
‘a cartoon’	‘art of a’
‘a sketch of the’	‘a pixelated photo of a’
‘itap of the’	‘a good photo of a’
‘a plushie’	‘a photo of the nice’
‘a photo of the small’	‘a photo of the weird’
’the cartoon’	‘art of the’
‘a drawing of the’	‘a photo of the large’
‘the plushie’	‘a dark photo of a’
‘itap of a’	‘graffiti of the’
‘a toy’	‘itap of my’
‘a photo of a cool’	‘a photo of a small’
‘a 3d object of the’	‘a 3d object of a’
‘a 3d face of a’	‘a 3d face of the’

Table 3. List of templates that our method uses for augmentation. The input text prompt is added to the end of each sentence template.