Text and Image Guided 3D Avatar Generation and Manipulation

Anonymous Authors[1]

Abstract

Latent space manipulation has recently become an interesting topic, which involves finding latent directions that can be used to manipulate images toward specific attributes. However, efficient generation and manipulation of 3D models remains a challenge. In this work, we propose a CLIP -based 3D model manipulation method that can manipulate both the shape and texture of the model. More specifically, we use a generative 3D face GAN and create a fully differentiable rendering pipeline for image and text-based manipulations using CLIP and ID losses for face model manipulation. Unlike previous approaches, our method can manipulate both shapes and textures and directly optimizes the 3D model instead of rendered images. We demonstrate the effectiveness of our approach with extensive experiments and comparisons.

1. Introduction

Generative models and Generative Adversarial Networks (GAN) for 2D vision has made striking progress in recent years with several breakthroughs such as Progressive GAN (Karras et al. (2018b))and StyleGAN (Karras et al. (2019; 2020)), enabling high-resolution and high-quality image generation across multiple domains. 3D vision and the field of 3D generation have made similarly remarkable advances, with the emergence of implicit surface and volume representations enabling encoding, reconstruction, and generation of detailed models of watertight surfaces without suffering from the limitations of using a 3D Euclidean grid or meshes with fixed topology. While these implicit representation-based approaches result in a learnable surface parameterization that is not limited in resolution, they often require coordinate sampling and non-differentiable point cloud and mesh generation, which is also time-consuming. Other works such as GANFit (Gecer et al. (2019)) and TBGAN (Gecer et al. (2020)) confine 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 the advances in 3D generation, how to control the results of 3D generative models remains an active research question. However, several methods have been proposed to leverage the learned latent space of image generating GANs for both unsupervised (see Voynov & Babenko (2020); Jahanian et al. (2019); Wold et al. (1987); Yüksel et al. (2021)) and supervised (see Goetschalckx et al. (2019); Shen et al. (2020); Abdal et al. (2021); Kocasari et al. (2021); Patashnik et al. (2021)) manipulation of the generated output.

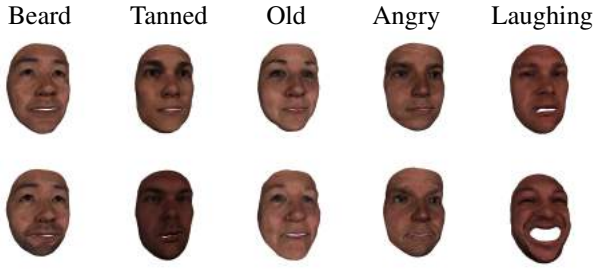

Figure 1. Example results of our method with various inputs and text prompts: "Beard", "Tanned", "Old", "Angry", "Laugh". The top row shows the original outputs, the bottom row shows the manipulation results, target text prompt is above each column.

In our work, we use a base generative model, TBGAN, that restricts 3D generation to 2D space by jointly generating 2D shape, normals and texture images, and draw insights from image manipulation problems to propose a fast and efficient text and image-guided mesh manipulation framework. Specifically, we first convert shape, normals, and texture images to a mesh, use a differentiable renderer (Kato et al. (2017); Liu et al. (2019)) to convert the mesh into an image, and leverage the joint image-text representational capabilities of CLIP (Radford et al. (2021)), a powerful multi-modal model, to directly optimize the shape, normals, and texture images. Unlike previous work that solely manipulates texture or the shape, our method can perform text and image-guided manipulations on both the shape and texture. We demonstrate that our method enables end-to-end differentiable rendering of high-quality and textured meshes and

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

enables both text and image-guided manipulations that can achieve fine-grained and complex manipulations such as changing the "gender", and "age" attributes without changing the irrelevant attributes or the identity of the original mesh. In short, our main contributions are as follows:

Fast text and image-guided manipulations: We propose a fast and efficient text and image-guided optimization pipeline that directly manipulate meshes to enhance or remove a variety of simple and complex attributes. Unlike previous work, our method takes only 4-5 minutes to achieve fine-grained and accurate manipulations.

Disentanglement of shape and texture manipulations: We use a low-resolution intermediate layer of TBGAN and propose a manipulation method that can perform shape-specific, texture-specific, or joint shape-texture manipulations without changing any irrelevant attributes.

Prompt engineering: We perform prompt engineering (see Radford et al. (2021)) to augment the input text by embedding it into text templates to achieve more accurate and consistent CLIP-based manipulations.

View augmentation: We generate multiple view renders of the generated meshes and compute a CLIP loss by averaging the distances between the embeddings of each rendered image and target text/image for manipulation. We show that view augmentation effectively prevents artifact generation.

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. Hence, work on 3D vision problems uses a wide variety of representations 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 that consist of coordinate values in (x, y, z) format, and are extensively used in 3D learning problems such as reconstruction of 3D shapes (Park et al., 2011), 3D object classification (Fan et al., 2017; Qi et al., 2017), and segmentation (Qi et al., 2017). However, point clouds provide no information on how the points are connected and pose view-dependency problems. Another 3D format, meshes, describes each shape as a set of triangular faces and connected vertices. While meshes are better suited to describe object topology, they often require advanced preprocessing steps to ensure an equal number of vertices and faces across all input data. 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 be able to describe fine-grained details. Finally, there have been a large number of neural implicit representations proposed in recent years to overcome the shortcomings of the classical representations. These methods represent 3D shapes as level sets of deep networks that map 3D coordinates to a signed distance function (SDF) (Park et al., 2019), or occupancy fields (Chen & Zhang, 2019; Mescheder et al., 2019), and aim to create a lightweight, continuous shape representation.

2.2. Latent Space Manipulation

Several methods have been proposed to exploit the latent space of GANs for image manipulation, which can be divided into two broad categories: supervised and unsupervised methods. Supervised approaches typically benefit from pre-trained attribute classifiers that guide the optimization process to discover meaningful directions in the latent space, or use labeled data to train new classifiers that directly aim to learn directions of interest (Goetschalckx et al., 2019; Shen et al., 2020). Other work shows that it is possible to find meaningful directions in latent space in an unsupervised way (Voynov & Babenko, 2020; Jahanian et al., 2019). GANSpace (Härkönen et al., 2020) proposes to apply Principal Component Analysis (PCA) (Wold et al., 1987) to randomly sample latent vectors of the intermediate layers of BigGAN and StyleGAN models. A similar approach is used in SeFA (Shen & Zhou, 2020), where they directly optimize the intermediate weight matrix of the GAN model in closed form. LatentCLR (Yüksel et al., 2021) proposes a contrastive learning approach to find unsupervised directions that are transferable to different classes. Moreover, both StyleCLIP (Patashnik et al. (2021)) and StyleMC (Kocasari et al. (2021)) use CLIP to find text-based directions within the latent spaces of StyleGAN2 to perform both coarse and fine-grained manipulations of various attributes. Another recent work (Abdal et al., 2021) proposes a method for attribute-conditioned sampling and attribute-controlled editing with StyleGAN2.

2.3. 3D Shape Generation and Manipulation

There have been tremendous advances in 3D shape generation in recent years. While some of these work use traditional 3D representations such as point clouds (Fan et al. (2017); Qi et al. (2017); Achlioptas et al. (2018); Hui et al. (2020); Shu et al. (2019)), voxels (Kar et al. (2015); Choy et al. (2016)) and meshes (Han et al. (2016); Hanocka et al. (2019)), there have been several approaches proposed to use implicit surface and volume representations for high-quality, efficient and scalable representations (see Park et al. (2019); Chen & Zhang (2019); Mescheder et al. (2019)). However, work on 3D shape manipulation is much more limited and focuses either on the manipulation of only shapes or textures. We broadly categorize work on 3D manipulation to two categories: unsupervised and supervised method.

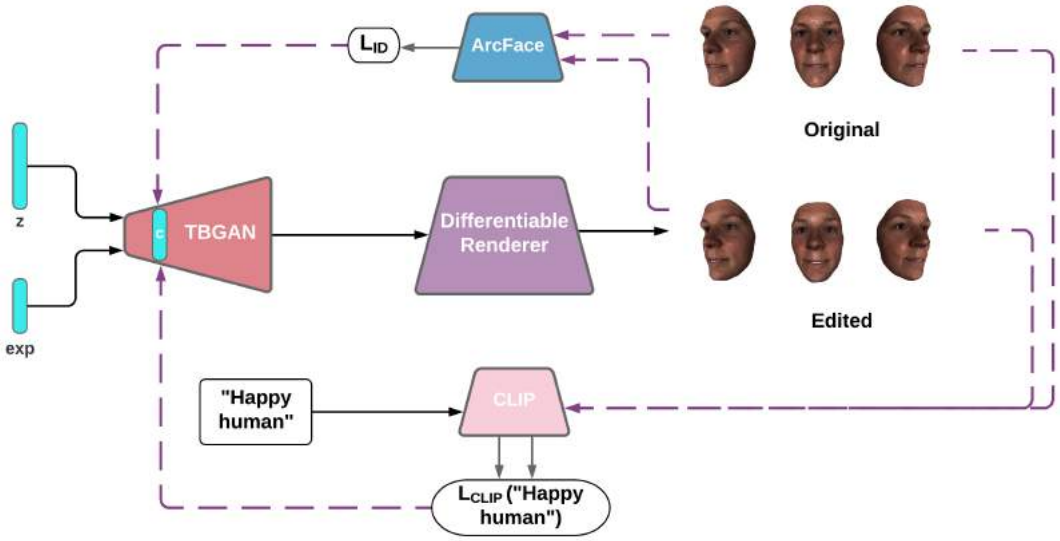

Figure 2. 3D-Style framework (using the text prompt *'Smiling'* as an example).The latent code $\mathbf{c}$ and $\Delta\mathbf{c} + \mathbf{c}$ are passed through the generator.The manipulation direction $\Delta\mathbf{s}$ corresponding to the text prompt is optimized by minimizing CLIP loss,identity loss and L2 loss.

Unsupervised 3D manipulation methods often introduce additional constraints into the training process to enforce better controllability. For example, SP-GAN Li et al. (2021) proposes a sphere-guided generative model to synthesize diverse and high-quality shapes in point cloud format while promoting controllability for part-aware shape manipulation. Similarly, Elsner et al. (2021) proposes an auto-encoder architecture with a Lipschitz-type constraint to the loss function in order to bound the change of the output shape proportionally to the change in latent space. This approach aims to encode a single shape and enable more intuitive shape editing.

Moreover, multiple supervised 3D manipulation methods have been proposed recently. Michel et al. (2021) proposes a neural style model that encodes the style of a single mesh and uses a CLIP-based method for color and local geometric detail manipulations. However, this method requires training a separate model for each shape to be manipulated. Another method, CLIP-Forge (Sanghi et al. (2021)), trains an auto-encoding occupancy network and a normalizing flow model to connect the CLIP encodings of 2D renders of 3D shapes and the latent shape encodings. While effective, this method is very limited in resolution due to voxel generation and takes more than 2 hours per manipulation.

3. Methodology

3.1. Background on TBGAN

In our work, we use TBGAN as our base generative model for manipulation, which is a GAN model with an architecture that closely resembles PG-GAN Karras et al. (2018a). More specifically, TBGAN proposes a progressively growing architecture that takes a one-hot-encoded expression vector and a random noise vector as input, progressively generates higher dimensional intermediate layer vectors, known as modality correlation layers, and branches out to modality-specific layers to jointly generate high-quality shape, shape normals and texture images. The model is trained on large-scale high-resolution UV-maps of pre-processed meshes with WGAN-GP loss Gulrajani et al. (2017). Within the modality correlation layers, the trunk network preserves the correspondences of modalities while the separate branches of modality-specific layers enable learning independent distributions for shape and texture data. Hence, the trained model enables disentangled manipulation of various coarse and fine-grained attributes such as facial features (i.e. "small eyes", "big lips"), overall facial shape (i.e. "rounded face", "asian") along with expression and personal identity (i.e. "beyonce"). Unlike 3DMMs, which are constructed by applying PCA on datasets of 3D scans of hundreds or thousands of subjects, TBGAN is not bound by linear separability constraints and provides a continuous latent space that can be easily manipulated.

3.2. Differentiable Rendering and Augmentation

Given a pre-trained TBGAN generator $\mathcal{G}$, let $z \in \mathcal{R}^d$ 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. Moreover, let $\mathcal{I}_s$, $\mathcal{I}_n$ and $\mathcal{I}_t$ denote the shape, normal and texture UV maps generated with TBGAN such that $\mathcal{G}(z, e) = (\mathcal{I}_s, \mathcal{I}_n, \mathcal{I}_t)$. Shape and

normal UV maps $\mathcal{I}_s$ and $\mathcal{I}_n$ store the 3D coordinates of each vertex (x, y, z) and the normal orientation (nx, ny, nz) respectively and can be directly projected back into 3D space. Our method takes the generated shape, normal and texture UV maps and generates a textured mesh $\mathcal{M}$ via spherical projection. Next, we use an open-source differentiable renderer library, PyTorch3D (see Ravi et al. (2020)), to render 3 views of the generated mesh from an anchor with the mesh $\mathcal{M}$ rotated -30, 3 and 30 degrees: $f_{diff}(\mathcal{M}, \theta_{cam}) = I$, where θ denotes the camera, object position and rotation parameters used by the differentiable renderer. We denote the renders generated by f_{diff} as I_1, I_2 and I_3. For text and image-based manipulation, we compute the CLIP loss for each image and average them prior to feeding them into our loss, which enforces view consistency and leads to more stable results. We note that we limit the number of views to 3 as TBGAN generates partial face meshes and not full head meshes. We describe the details of our text and image-guided manipulation method in Section 3.3.

3.3. Text and Image Guided Manipulation

Let $c \in \mathcal{C}$ denote an intermediate layer vector that is obtained by partial forward propagation of z and e through the generator $\mathcal{G}$. Our method takes a text prompt t such as '*A woman with big lips*' or '*A man with beard*' as input and finds a manipulation $\Delta\mathbf{c}$ such that $G(\mathbf{c} + \Delta\mathbf{c})$ generates a manipulated image in which the target attribute specified by t is present or enhanced, while other attributes remain mostly unaffected. A diagram of our method is shown in Figure 2.

To perform meaningful manipulation of meshes without generation artifacts or changing irrelevant attributes, we use a combination of a CLIP loss, identity loss, and L2 loss. We use differentiable rendering to capitalize on the joint representational power of CLIP and use rendered images to directly optimize the 3D generation output. As described in Section 3.2, we generate three renders with different views at each optimization step for more stable optimization. Furthermore, we augment the original text prompt t by embedding it into sentence templates to generate k text prompts $t_1, t_2, ..., t_k$. We compute a CLIP-based loss term $\mathcal{L}_{CLIP}$ to minimize the cosine distance between CLIP embeddings of the rendered images $I_i$and the set of text prompts t_j. Furthermore, we propose an alternative image-guided manipulation method where the CLIP-based loss term $\mathcal{L}_{CLIP}$ seeks to minimize the cosine distance between CLIP embeddings of the rendered images $I_i$and the target image I_{targ}. The two alternative CLIP-based losses are as follows:

$$\mathcal{L}_{CLIP} = \frac{\sum_{j=1}^{k}\sum_{i=1}^{3} D_{CLIP}(I_i, t_j)}{k * 3} \tag{1}$$

$$\mathcal{L}_{CLIP} = \frac{\sum_{i=1}^{3} D_{CLIP}(I_i, I_{targ})}{3} \tag{2}$$

where I_i is ith rendered image, t_j is the target text t embedded into the jth text template, I_{targ} is the target image (in the case of image-guided manipulation) and D_{CLIP} is the cosine distance between CLIP embeddings. We also use an identity loss that minimizes the distance between the identity of the original renders and manipulated renders, and an L2 loss to prevent artifact generation:

$$\mathcal{L}_{ID} = 1 - \langle R(G(c)), R(G(c + \Delta c))\rangle \tag{3}$$

$$\mathcal{L}_{L2} = ||c - (c + \Delta c)||_2 \tag{4}$$

where R is ArcFace (Deng et al., 2019), a facial recognition network in the case of face recognition, and $\langle \cdot, \cdot \rangle$ computes cosine similarity between the identities of unperturbed rendered image and manipulated result. We find that using identity loss effectively prevents changes to irrelevant attributes. The compound loss of our network is formulated as follows:

$$\underset{\Delta c \in\ 4x4/Dense}{arg\,min} \quad \mathcal{L}_{CLIP} + \lambda_{ID}\mathcal{L}_{ID} + \lambda_{L2}\mathcal{L}_{L2} \tag{5}$$

Where λ_{ID} and λ_{L2} are manipulation strength hyperparameters of $\mathcal{L}_{ID}$ and $\mathcal{L}_{L2}$, respectively. While the CLIP loss ensures that the attribute specified by the text prompt is enhanced, ID loss and L2 loss keep other attributes unchanged, thus enforcing disentangled changes. In our work, we optimize the original intermediate latent vector c via gradient descent and operate in the 4x4/Dense layer of TBGAN generator. The optimized latent vector $c + \Delta c$ then can be fed into TBGAN to generate shape, normal, and texture UV maps and finally a manipulated mesh with the target attributes.

4. Experiments

In this section, we present the experimental results of our proposed manipulation method and evaluate our results based on manipulation quality and fidelity. We examine our method across a diverse set of generated face meshes and demonstrate the effectiveness of our approach in achieving both coarse and fine-grained manipulations. Moreover, we present comparisons with baseline manipulation methods, PCA, and show that our method enables more disentangled manipulation without changing the identity of the manipulated avatar. Finally, we conducted a human evaluation to demonstrate the effectiveness of our approach at achieving the target manipulation and present additional human evaluation to measure generation quality.

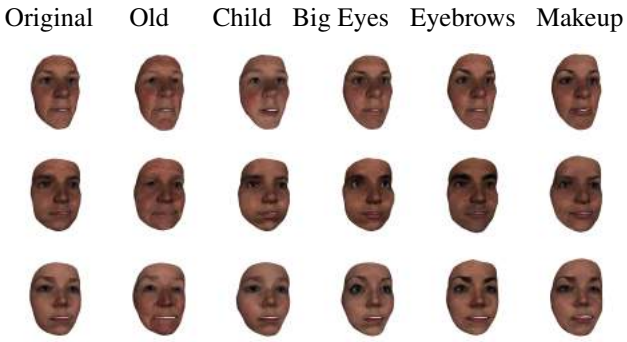

Figure 3. Manipulation results of our method with various inputs and text prompts on three different faces: "Old", "Child", "Big Eyes", "Thick Eyebrows", "Makeup". Leftmost column show the original outputs, adjacent columns show the manipulation results, target text prompt is above each column.

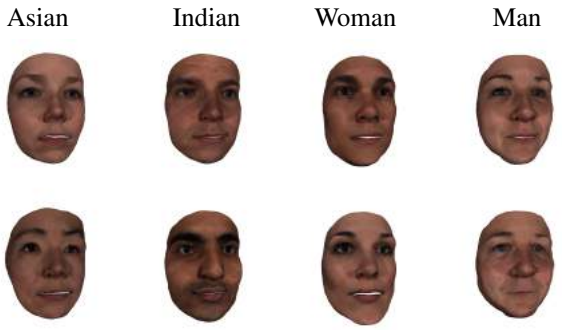

Figure 4. Manipulation results of our method with various inputs and text prompts: "Asian", "Indian", "Woman", "Man". The top row shows the original outputs, the bottom row shows the manipulation results, target text prompt is above each column.

4.1. Experimental Setup

We use the official implementation of TBGAN [1] and the trained TBGAN model provided by the authors. For all manipulation experiments, we start by generating random meshes with TBGAN with the default hyperparameters. For differentiable rendering, we use the renderer of PyTorch3D without additional training or finetuning 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 point lights location is taken to be $(0.0,\ 0.0,\ +3.0)$.

Moreover, we use $10,000$ randomly generated latent vectors extracted from the 4x4/Dense layer of TBGAN to perform PCA and find unsupervised directions. We apply the found directions to previously unseen, randomly generated latent vectors and manually inspect the manipulation results. For all manipulation experiments with CLIP, we set the loss terms as follows: $\lambda_{ID} = 0.01$, $\lambda_{L2} = 0.0001$. We note that a single optimization step consisting of mesh generation, differentiable rendering, and loss calculation with CLIP takes 4 minutes. We use a fixed number of 100 optimization steps for each manipulation. Moreover, we use the Adam optimizer of Tensorflow and keep the default hyperparameters. All experiments are performed on a single TITAN RTX GPU.

4.2. Manipulation Quality

We demonstrate the quality and consistency of the manipulations performed by our method on a diverse set of generated faces. We start by performing a set of text-based manipulations that range from coarse and simple attributes such as "big eyes", "thick eyebrows", "makeup" to fine-grained and complex attributes such as "age", and present the results in Figure 3. As can be seen in the figure, our method successfully performs the target manipulations on different faces and generates details with high granularity while still maintaining global semantics and preserving the underlying content. For example in Figure 3, given generated shapes of different faces and target text prompt "old", the manipulated outputs depict old people, while preserving the overall shape or identity of the original meshes.

Next, we investigate whether our base model TBGAN, combined with our CLIP-based manipulation method demonstrates a global semantic understanding of more complex human attributes such as ethnicity and gender. We note that these attributes are not only interesting due to their complexity but because they are often significant sources of bias in generative models. To this end, we perform "man", "woman", "Asian", and "Indian" manipulations on various randomly generated outputs, and present the manipulation results in Figure 4.

The manipulation capabilities of our method are not limited to the physical characteristics of generated avatars but can be used to change their facial expressions such as "smiling", "angry", and "surprised". We note that the ability to change the expressions of the avatars is especially important in the case of our base model as it enables sequential manipulations such that a mesh can be manipulated in the "beard" direction first and then in the "angry" direction. As shown in Figure 5, our method can achieve successful edits for a variety of complex emotions on various input meshes with almost no change to other attributes. Moreover, our model is capable of controlling the intensity of the expression by increasing and decreasing the $\mathcal{L}_{ID}$. As $\mathcal{L}_{ID}$ increases the level of the expression decreases as can be seen in 6.

In addition to general physical attribute and expression manipulations, we demonstrate that our method can be used

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

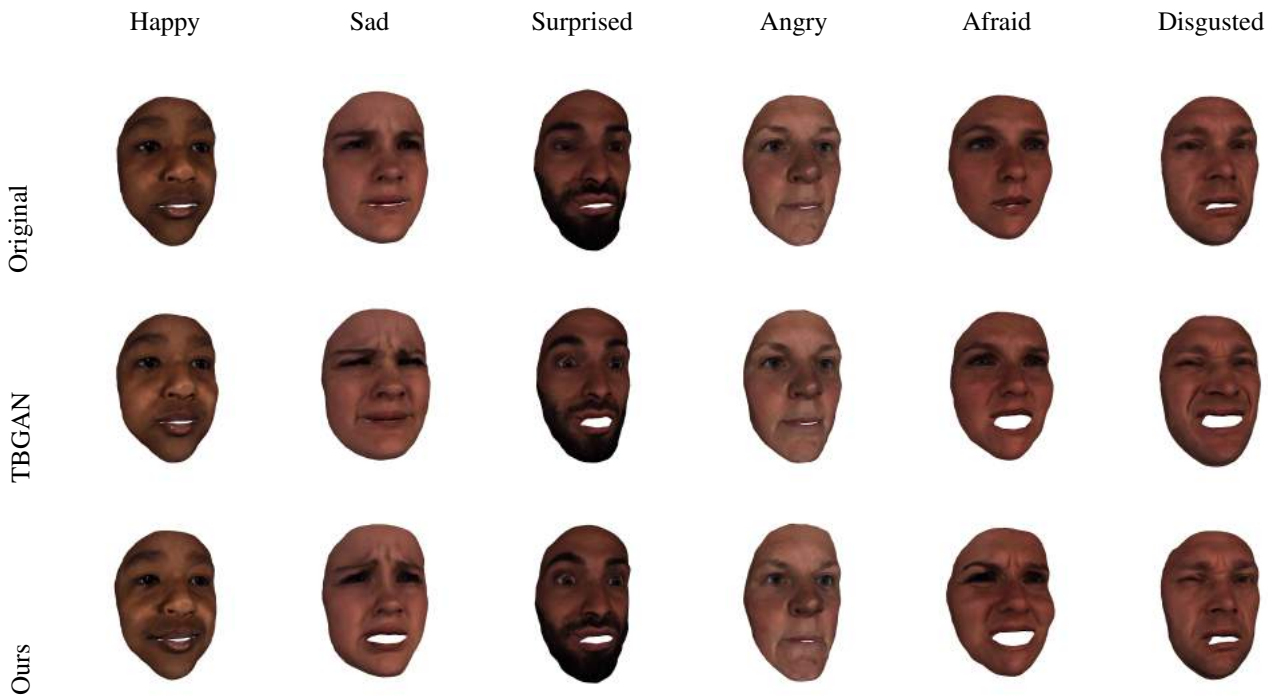

Figure 5. Manipulation results of our method with various inputs and text prompts: "Happy", "Sad", "Surprised", "Angry", "Afraid", and "Disgusted" manipulations. The top row shows the original outputs, the middle row shows the TBGAN conditioned expressions, the bottom row shows the manipulation results, target text prompt is above each column.

Figure 6. The results for different manipulation strengths on "sad" text prompt. From left to right, $\mathcal{L}_{ID}$ decreases, and the "sadness" increases.

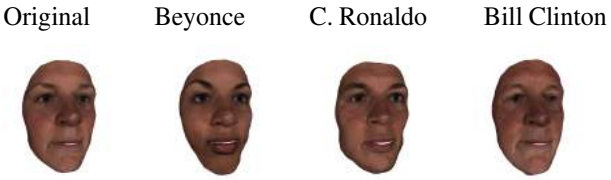

Figure 7. Results of "Beyonće", "Cristiano Ronaldo", and "Bill Clinton" manipulations with our method. The leftmost shows the original input, the ones next to it show the manipulation results.

to perform complex identity manipulations with solely text-based manipulations. For this experiment, we perform "Bill Clinton", "Cristiano Ronaldo", and "Beyonce" manipulations on random generation outputs and show the results in Figure 7. As shown in the Figure, our method is able to achieve a manipulation that captures the characteristics of the target person, such as the characteristic drooped eyes and slightly caved in the forehead of Bill Clinton.

Finally, we provide a qualitative comparison of our method with a PCA-based baseline. For this experiment, we sample $10,000$ intermediate latent vectors from the 4x4/Dense layer of TBGAN and apply PCA on the concatenated latent vectors to obtain principal components with each component representing a new transformed axis that is a linear combination of original features. The attributes encoded by each principal component are a combination of the original features and require a manual inspection to be interpreted. 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:

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

Here α denotes the step size, k denotes the number of steps, and PC_i is the i_{th} principal component (direction) used. For comparison purposes, we keep the highest ranked 5 principal components and apply them randomly generated latent

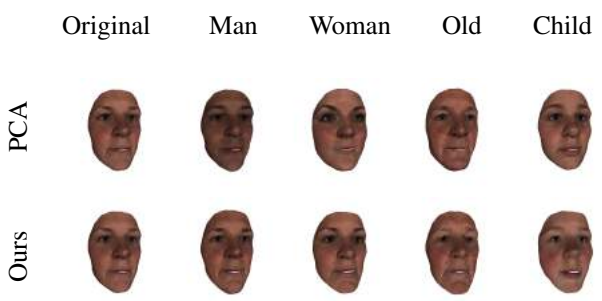

Figure 8. Comparison between PCA-based manipulation and text-guided manipulation with our method. The top row shows PCA-based manipulations, bottom row show manipulations with our method.

vectors with step size α set to 10. Moreover, we observe that the top components encode very prominent directions: age and gender. For comparison, we apply age and gender-related text-guided edits on the same latent vectors and present the comparative results in Figure 8. As can be seen in the figure, the top directions encoded by PCA, age, and gender, significantly change the identity of the input person, whereas our method achieves the desired manipulations without changing any irrelevant attributes.

4.3. Image-Guided Manipulation

We note that our method is not limited to text-guided manipulation but can be used to manipulate a mesh with different target modalities such as a 2D face image as described in Section 3.3. For a target 2D image I_{targ}, our method seeks to generate rendered images I_i. We note that, unlike text-guided manipulation, image-based manipulations inevitably change both coarse and fine-grained attributes and is better suited for general use cases where identity preservation is not as important. In lieu with the use case, we set the loss terms $\lambda_{ID} = 0.01$ and $\lambda_{L2} = 0$, and perform manipulations with various face images. Figure 9 shows the manipulation results for various target images of celebrities. As seen in the figure, our method can capture complex identity-related attributes of the target image regardless of illumination and pose, and can perform successful manipulations.

4.4. Effect of Identity Loss

Our method uses ArcFace, a large-scale pre-trained face recognition network, to compute an identity loss $\mathcal{L}_{ID}$ and enforce identity preservation during manipulation. Therefore, we perform an ablation study with various input meshes and various target texts describing emotion, shape, and texture related changes to demonstrate the effect of $\mathcal{L}_{ID}$ on manipulation output, and present the results in Figure 10.

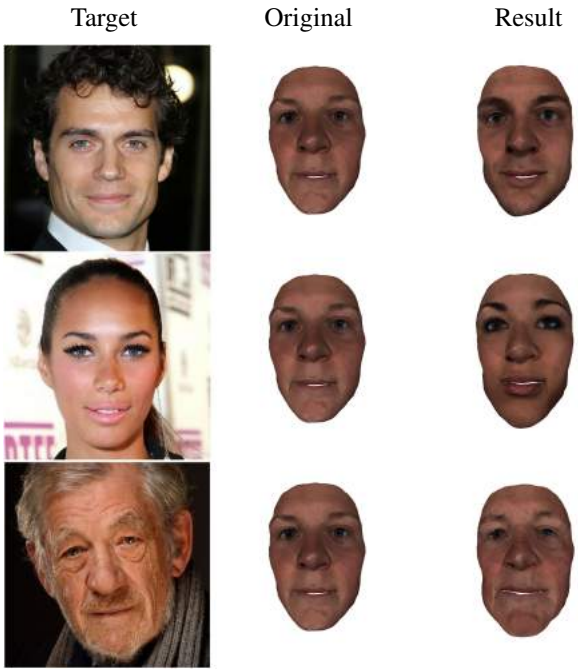

Figure 9. Results of manipulations with our image-guided method.

For the identity loss experiments, we simply set $\mathcal{L}_{ID} = 0$ and keep other hyperparameters the same. As can be seen in Figure 10, the identity loss is crucial to preserving the identity of the input, and omitting it yields manipulation results that are vastly different from the input.

4.5. Human Evaluation

Our method performs text-driven manipulation of meshes. Given that no approaches exist for this task, we evaluate our method's performance via human evaluation to evaluate the perceived quality of the generated outputs. More specifically, we are interested in how much the manipulation preserves the attributes irrelevant to the target manipulation (when relevant) and how well the manipulation result matches the target text input.

For human evaluation, we asked 25 users to evaluate 5 randomly selected sets of input meshes, text prompts t, and output/manipulated meshes. For each set, we display the target text and the output in pairs, and ask users to assign a score between $[1, 5]$ for two questions: "Is the identity of the input face preserved?" "How well does the manipulation achieve the attributes specified in the target text?". We report the mean scores and their standard deviations of questions for our method and the PCA-based baseline in Table 1. Also, we conducted a survey to compare our method's performance on expressions with TBGAN's expressions. The structure of this survey was the same as before and this time

	Maleness	Oldness	Femaleness	Youngness	All
PCA	$\mathbf{4.42}(\pm 0.65)$	$\mathbf{4.25}(\pm 0.61)$	$\mathbf{4.13}(\pm 0.99)$	$\mathbf{4.21}(\pm 0.83)$	$\mathbf{4.25}(\pm 0.78)$
Ours	$\mathbf{4.21}(\pm 0.83)$	$\mathbf{4.67}(\pm 0.56)$	$\mathbf{4.08}(\pm 0.97)$	$\mathbf{4.46}(\pm 0.78)$	$\mathbf{4.35}(\pm 0.82)$
PCA+Identity	$\mathbf{2.85}(\pm 1.04)$	$\mathbf{3.52}(\pm 1.29)$	$\mathbf{2.57}(\pm 1.08)$	$\mathbf{3.01}(\pm 1.30)$	$\mathbf{2.99}(\pm 1.21)$
Ours+Identity	$\mathbf{4.45}(\pm 0.76)$	$\mathbf{3.70}(\pm 1.22)$	$\mathbf{4.33}(\pm 0.80)$	$\mathbf{3.62}(\pm 1.20)$	$\mathbf{4.02}(\pm 1.07)$

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

	Surprise	Disgust	Happy	Sad	Afraid	Angry	All
TBGAN	$\mathbf{4.64}(\pm 0.48)$	$\mathbf{4.04}(\pm 1.12)$	$\mathbf{1.83}(\pm 0.92)$	$\mathbf{4.17}(\pm 0.87)$	$\mathbf{2.87}(\pm 1.08)$	$\mathbf{2.05}(\pm 1.22)$	$\mathbf{3.26}(\pm 1.46)$
Ours	$\mathbf{4.70}(\pm 0.46)$	$\mathbf{4.26}(\pm 0, 61)$	$\mathbf{3.77}(\pm 0.90)$	$\mathbf{4.48}(\pm 0.71)$	$\mathbf{3.43}(\pm 0.97)$	$\mathbf{3.65}(\pm 1.24)$	$\mathbf{4.05}(\pm 0.97)$

Table 2. Mean scores (1-5) for manipulation accuracy for our method's and TBGAN's expressions.

Figure 10. Results with (left) and without identity loss (right). From top to bottom the order of manipulations is original, child, makeup, big eyes.

we asked one question: "How well does the manipulation achieve the expressions specified in the target text?". We report the mean scores and their standard deviations of the question for our method and the TBGAN in Table 2.

As the results of the human evaluation show, our method performs slightly better than PCA in all settings except Female and Male attributes. However, we note that our method outperforms PCA significantly for these attributes for identity preservation rated by human evaluators. More specifically, the human evaluators found that our method preserves identity while achieving competitive scores to attribute preservation by $\%34$ higher than PCA.

Moreover, our method outperforms TBGAN in all settings.

5. Limitations and Broader Impact

While our method is highly effective at achieving both coarse and fine-grained manipulations, our base generative model is trained to generate partial face avatars. Hence, we strongly believe that our work can be extended to more complete generation scenarios to generate full head meshes and/or bodies.

6. Conclusion

We have proposed an approach to manipulate 3D avatars with text and image inputs that relies on a differentiable rendering pipeline combined with CLIP-based and identity-based losses. Our method leverages the joint representational capabilities of CLIP and operates within the low-resolution layers of a base generative model, TBGAN, to find a manipulation that can achieve the attributes specified in the target text/image without changing irrelevant attributes. Unlike previous work, which confine 3D shape manipulation to either local geometric changes such as texture or solely the shape, our method can perform high-quality and complex edits of shapes and textures both separately and jointly. Another significant advantage of our method is it takes only 4-5 minutes to manipulate a given mesh, while other works require either hour of optimization time per manipulation.

We note that avatar and human body generation, in general, have extensive use cases in the industry, such as character design, animation, and visual effects. Hence, we can see two natural improvements to our work: sketch-based manipulation for more intuitive and user-friendly manipulations and extension of our framework to full body generation.

References

Abdal, R., Zhu, P., Mitra, N. J., and Wonka, P. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ArXiv*, abs/2008.02401, 2021.

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. J. Learning representations and generative models for 3d point clouds. In *ICML*, 2018.

Chen, Z. and Zhang, H. Learning implicit fields for gener-

ative shape modeling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5932–5941, 2019.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pp. 628–644. Springer, 2016.

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

Elsner, T., Ibing, M., Czech, V., Nehring-Wirxel, J., and Kobbelt, L. P. Intuitive shape editing in latent space. *ArXiv*, abs/2111.12488, 2021.

Fan, H., Su, H., and Guibas, L. J. 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*, pp. 605–613, 2017.

Gecer, B., Ploumpis, S., Kotsia, I., and Zafeiriou, S. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1155–1164, 2019.

Gecer, B., Lattas, A., Ploumpis, S., Deng, J., Papaioannou, A., Moschoglou, S., and Zafeiriou, S. Synthesizing coupled 3d face modalities by trunk-branch generative adversarial networks. In *European Conference on Computer Vision*, pp. 415–433. Springer, 2020.

Goetschalckx, L., Andonian, A., Oliva, A., and Isola, P. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5744–5753, 2019.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *NIPS*, 2017.

Han, Z., Liu, Z., Han, J., Vong, C.-M., Bu, S., and Chen, C. L. P. 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.

Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., and Cohen-Or, D. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38:1 – 12, 2019.

Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

Hui, L., Xu, R., Xie, J., Qian, J., and Yang, J. Progressive point cloud deconvolution generation network. In *ECCV*, 2020.

Jahanian, A., Chai, L., and Isola, P. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Kar, A., Tulsiani, S., Carreira, J., and Malik, J. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1966–1974, 2015.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2018a.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation, 2018b.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2020.

Kato, H., Ushiku, Y., and Harada, T. Neural 3d mesh renderer, 2017.

Kocasari, U., Dirik, A., Tiftikci, M., and Yanardag, P. Stylemc: Multi-channel based fast text-guided image generation and manipulation. *ArXiv*, abs/2112.08493, 2021.

Li, R., Li, X., Hui, K.-H., and Fu, C.-W. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ArXiv*, abs/2108.04476, 2021.

Liu, S., Li, T., Chen, W., and Li, H. Soft rasterizer: A differentiable renderer for image-based 3d reasoning, 2019.

Mescheder, L. M., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4455–4465, 2019.

Michel, O. J., Bar-On, R., Liu, R., Benaim, S., and Hanocka, R. Text2mesh: Text-driven neural stylization for meshes. *ArXiv*, abs/2112.03221, 2021.

Park, J., Kim, H., Tai, Y.-W., Brown, M. S., and Kweon, I. High quality depth map upsampling for 3d-tof cameras. In *2011 International Conference on Computer Vision*, pp. 1623–1630. IEEE, 2011.

Park, J. J., Florence, P. R., Straub, J., Newcombe, R. A., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174, 2019.

Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Ravi, N., Reizenstein, J., Novotný, D., Gordon, T., Lo, W.-Y., Johnson, J., and Gkioxari, G. Accelerating 3d deep learning with pytorch3d. *SIGGRAPH Asia 2020 Courses*, 2020.

Sanghi, A., Chu, H., Lambourne, J., Wang, Y., Cheng, C.-Y., and Fumero, M. Clip-forge: Towards zero-shot text-to-shape generation. *ArXiv*, abs/2110.02624, 2021.

Shen, Y. and Zhou, B. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

Shen, Y., Yang, C., Tang, X., and Zhou, B. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Shu, D. W., Park, S. W., and Kwon, J. 3d point cloud generative adversarial network based on tree structured graph convolutions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3858–3867, 2019.

Voynov, A. and Babenko, A. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pp. 9786–9796. PMLR, 2020.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Yüksel, O. K., Simsar, E., Er, E. G., and Yanardag, P. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14263–14272, October 2021.