

# FaceCLIPNeRF: Text-driven 3D Face Manipulation using Deformable Neural Radiance Fields

Sungwon Hwang<sup>1</sup> Junha Hyung<sup>1</sup> Daejin Kim<sup>2</sup> Min-Jung Kim<sup>1</sup> Jaegul Choo<sup>1</sup>

<sup>1</sup>KAIST <sup>2</sup>Scatter Lab

{shwang.14, sharpeeee, emjay73, jchoo}@kaist.ac.kr, daejin@scatterlab.co.kr

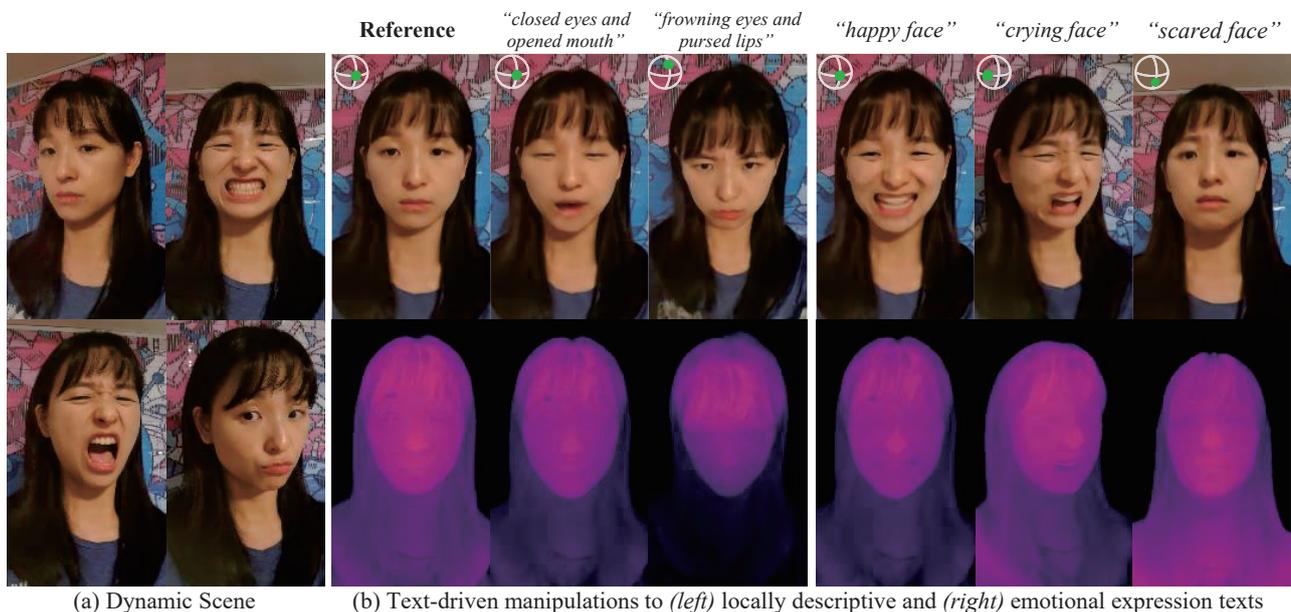


Figure 1: **FaceCLIPNeRF** reconstructs a video of a dynamic scene of a face, and conducts face manipulation using texts only. Manipulated faces and their depths in top and bottom rows in (b), respectively, are rendered from novel views.

## Abstract

As recent advances in Neural Radiance Fields (NeRF) have enabled high-fidelity 3D face reconstruction and novel view synthesis, its manipulation also became an essential task in 3D vision. However, existing manipulation methods require extensive human labor, such as a user-provided semantic mask and manual attribute search unsuitable for non-expert users. Instead, our approach is designed to require a single text to manipulate a face reconstructed with NeRF. To do so, we first train a scene manipulator, a latent code-conditional deformable NeRF, over a dynamic scene to control a face deformation using the latent code. However, representing a scene deformation with a single latent code is unfavorable for compositing local deformations observed in different instances. As so, our proposed Position-conditional Anchor Compositor (PAC) learns to represent

a manipulated scene with spatially varying latent codes. Their renderings with the scene manipulator are then optimized to yield high cosine similarity to a target text in CLIP embedding space for text-driven manipulation. To the best of our knowledge, our approach is the first to address the text-driven manipulation of a face reconstructed with NeRF. Extensive results, comparisons, and ablation studies demonstrate the effectiveness of our approach.

## 1. Introduction

Easy manipulation of 3D face representation is an essential aspect of advancements in 3D digital human contents[33]. Though Neural Radiance Field[21] (NeRF) made a big step forward in a 3D scene reconstruction, many of its manipulative methods targets color[4, 35] or rigid ge-

ometry [46, 16, 42, 15] manipulations, which are inappropriate for detailed facial expression editing tasks. While a recent work proposed a regionally controllable face editing method [13], it requires an exhaustive process of collecting user-annotated masks of face parts from curated training frames, followed by manual attribute control to achieve a desired manipulation. Face-specific implicit representation methods [6, 48] utilize parameters of morphable face models [37] as priors to encode observed facial expressions with high fidelity. However, their manipulations are not only done manually but also require extensive training sets of approximately 6000 frames that cover various facial expressions, which are laborious in both data collection and manipulation phases. On the contrary, our approach only uses a single text to conduct facial manipulations in NeRF, and trains over a dynamic portrait video with approximately 300 training frames that include a few types of facial deformation examples as in Fig. 1a.

In order to control a face deformation, our method first learns and separates observed deformations from a canonical space leveraging HyperNeRF[24]. Specifically, per-frame deformation latent codes and a shared latent code-conditional implicit scene network are trained over the training frames. Our key insight is to represent the deformations of a scene with multiple, spatially-varying latent codes for manipulation tasks. The insight originates from the shortcomings of naively adopting the formulations of HyperNeRF to manipulation tasks, which is to search for a single latent code that represents a desired face deformation. For instance, a facial expression that requires a combination of local deformations observed in different instances is not expressible with a single latent code. In this work, we define such a problem as “*linked local attribute problem*” and address this issue by representing a manipulated scene with spatially varying latent codes. As a result, our manipulation could express a combination of locally observed deformations as seen from the image rendering highlighted with red boundary in Fig. 2a.

To this end, we first summarize all observed deformations as a set of anchor codes and let MLP learn to compose the anchor codes to yield multiple, position-conditional latent codes. The reflectivity of the latent codes on visual attributes of a target text is then achieved by optimizing the rendered images of the latent codes to be close to a target text in CLIP[28] embedding space. In summary, our work makes the following contributions:

- Proposal of a text-driven manipulation pipeline of a face reconstructed with NeRF.
- Design of a manipulation network that learns to represent a scene with spatially varying latent codes.
- First to conduct text-driven manipulation of a face reconstructed with NeRF to the best of our knowledge.

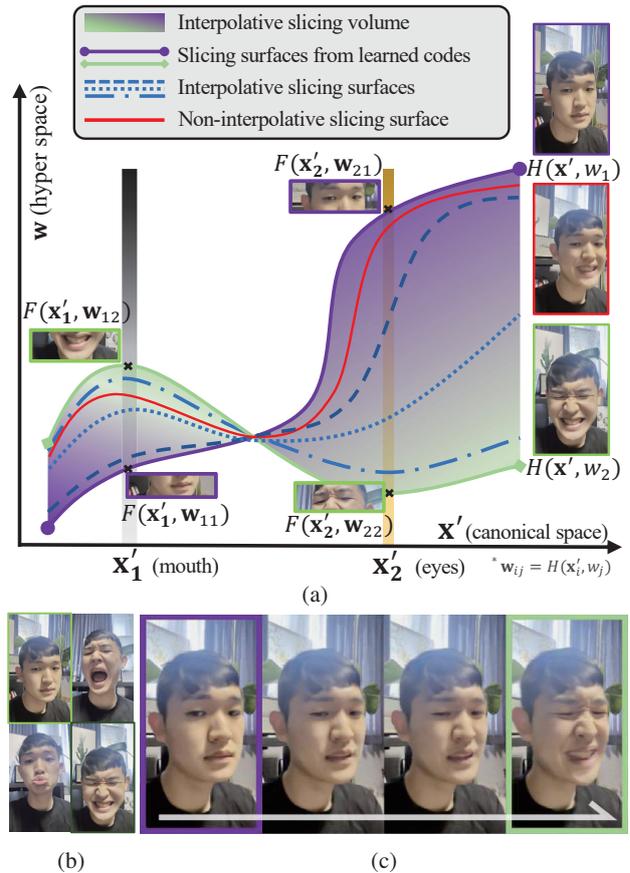


Figure 2: (a) Illustration of linked local attribute problem in hyper space. Expressing scene deformation with per-scene latent code cannot compose local facial deformation observed in different instances. (b) Types of facial deformations observed during scene manipulator training. (c) Renderings of interpolated latent codes with a scene manipulator.

## 2. Related Works

**NeRF and Deformable NeRF** Given multiple images taken from different views of a target scene, NeRF[21] synthesizes realistic novel view images with high fidelity by using an implicit volumetric scene function and volumetric rendering scheme[12], which inspired many follow-ups [1, 36, 20, 38, 45]. As NeRF assumes a static scene, recent works [23, 24, 27, 17] propose methods to encode dynamic scenes of interest. The common scheme of the works is to train a latent code per training frame and a single latent-conditional NeRF model shared by all trained latent codes to handle scene deformations. Our work builds on this design choice to learn and separate the observed deformations from a canonical space, yet overcome its limitation during the manipulation stage by representing a manipulated scene with spatially varying latent codes.

**Text-driven 3D Generation and Manipulation** Many works have used text for images or 3D manipulation[39, 9, 26, 11, 30, 10]. CLIP-NeRF[39] proposed a disentangled conditional NeRF architecture in a generative formulation supervised by text embedding in CLIP[28] space, and conducted text-and-exemplar driven editing over shape and appearance of an object. Dreamfields [9] performed generative text-to-3D synthesis by supervising its generations in CLIP embedding space to a generation text. We extend from these lines of research to initiate CLIP-driven manipulation of face reconstructed with NeRF.

**NeRF Manipulations** Among many works that studied NeRF manipulations[19, 46, 37, 13, 35, 34, 7, 50, 16], EditNeRF[19] train conditional NeRF on a shape category to learn implicit semantics of the shape parts without explicit supervision. Then, its manipulation process propagates user-provided scribbles to appropriate object regions for editing. NeRF-Editing[46] extracts mesh from trained NeRF and lets the user perform the mesh deformation. A novel view of the edited scene can be synthesized without re-training the network by bending corresponding rays. CoNeRF[13] trains controllable neural radiance fields using user-provided mask annotations of facial regions so that the user can control desired attributes within the region. However, such methods require laborious annotations and manual editing processes, whereas our method requires only a single text for detailed manipulation of faces.

**Neural Face Models** Several works[43, 29, 48] built 3D facial models using neural implicit shape representation. Of the works, i3DMM[43] disentangles face identity, hairstyle, and expression, making decoupled components to be manually editable. Face representation works based on NeRF have also been exploited[40, 37, 48]. Wang *et al.*[40] proposed compositional 3D representation for photo-realistic rendering of a human face, yet requires guidance images to extract implicitly controllable codes for facial expression manipulation. NerFACE[37] and IMavatar[48] model the appearance and dynamics of a human face using learned 3D Morphable Model[2] parameters as priors to achieve controllability over pose and expressions. However, the methods require a large number of training frames that cover many facial expression examples and manual adjustment of the priors for manipulation tasks.

### 3. Preliminaries

#### 3.1. NeRF

NeRF [21] is an implicit representation of geometry and color of a space using MLP. Specifically, given a point coordinate  $\mathbf{x} = (x, y, z)$  and a viewing direction  $\mathbf{d}$ , an MLP function  $\mathcal{F}$  is trained to yield density and color of the point as  $(\mathbf{c}, \sigma) = \mathcal{F}(\mathbf{x}, \mathbf{d})$ .  $M$  number of points are sampled along

a ray  $\mathbf{r} = \mathbf{o} + t\mathbf{d}$  using distances,  $\{t_i\}_{i=0}^M$ , that are collected from stratified sampling method.  $F$  predicts color and density of each point, all of which are then rendered to predict pixel color of the ray from which it was originated as

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^M T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i, \quad (1)$$

where  $\delta_i = t_{i+1} - t_i$ , and  $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j)$  is an accumulated transmittance.  $\mathcal{F}$  is then trained to minimize the rendering loss supervised with correspondingly known pixel colors.

#### 3.2. HyperNeRF

Unlike NeRF that is designed for a static scene, HyperNeRF [24] is able to encode highly dynamic scenes with large topological variations. Its key idea is to project points to canonical hyperspace for interpretation. Specifically, given a latent code  $w$ , a spatial deformation field  $T$  maps a point to a canonical space, and a slicing surface field  $H$  determines the interpretation of the point for a template NeRF  $F$ . Specifically,

$$\mathbf{x}' = T(\mathbf{x}, w), \quad (2)$$

$$\mathbf{w} = H(\mathbf{x}, w), \quad (3)$$

$$(\mathbf{c}, \sigma) = F(\mathbf{x}', \mathbf{w}, \mathbf{d}), \quad (4)$$

where  $w \leftarrow w_n \in \{w_1 \cdots w_N\} = W$  is a trainable per-frame latent code that corresponds to each  $N$  number of training frames. Then, the rendering loss is finally defined as

$$\mathcal{L}_c = \sum_{\substack{n \in \{1 \cdots N\}, \\ \mathbf{r}^n \in \mathcal{R}^n}} \|C_n(\mathbf{r}^n) - \hat{C}_n(\mathbf{r}^n)\|_2^2, \quad (5)$$

where  $C_n(\mathbf{r}^n)$  is ground truth color at  $n$ -th training frame of a ray  $\mathbf{r}^n$  and  $\mathcal{R}^n$  is a set of rays from  $n$ -th camera. Note that  $(\mathbf{x}', \mathbf{w})$  and  $H(\mathbf{x}, w)$  are often referred to *canonical hyperspace* and *slicing surface*, since  $\mathbf{x}'$  can be interpreted differently for different  $w$  as illustrated in Fig. 2a.

### 4. Proposed Method

We aim to manipulate a face reconstructed with NeRF given a target text that represents a desired facial expressions for manipulation (e.g., “crying face”, “wink eyes and smiling mouth”). To this end, our proposed method first trains a scene manipulator, a latent code-conditional neural field that controls facial deformations using its latent code (§4.1). Then, we elaborate over the pipeline to utilize a target text for manipulation (§4.2), followed by proposing an MLP network that learns to appropriately use the learned deformations and the scene manipulator to render scenes with faces that reflect the attributes of target texts (§4.3).

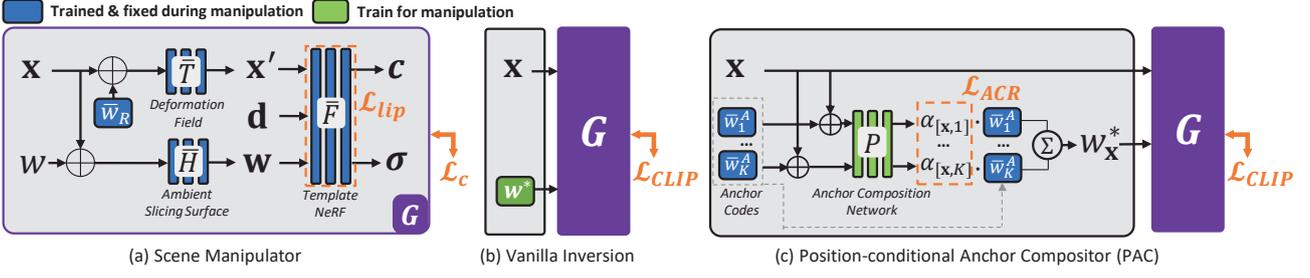


Figure 3: (a) Network structure of scene manipulator  $G$ . (b) Vanilla inversion method for manipulation. (c) Position-conditional Anchor Composer (PAC) for manipulation.

#### 4.1. Scene Manipulator

First, we construct a scene manipulator using HyperNeRF[24] so that deformations of a scene can be controlled by fixing the parameters of the scene manipulator and manipulating its latent code. Specifically, we train a dynamic scene of interest with a network formulated as Eq.(4) following [24], after which we freeze the trained parameters of  $T$ ,  $H$ ,  $F$ , and  $W$  and use  $w$  as a manipulation handle. In addition, we empirically found that the deformation network  $T$  tends to learn rigid deformations, such as head pose, while slicing surface field  $H$  learns non-rigid and detailed deformations, such as shapes of mouth and eyes. As so, we select and fix a trained latent code for  $T$  and only manipulate a latent code fed to  $H$ . In summary, as illustrated in Fig. 3(a), our latent code-conditional scene manipulator  $G$  is defined as

$$G(\mathbf{x}, \mathbf{d}, w) := \bar{F}(\bar{T}(\mathbf{x}, \bar{w}_R), \bar{H}(\mathbf{x}, w), \mathbf{d}), \quad (6)$$

where  $\bar{\cdot}$  represents that the parameters are trained and fixed for manipulation, and  $\bar{w}_R$  is a fixed latent code of the desired head pose chosen from a set of learned latent codes  $\bar{W}$ . In the supplementary material, we report further experimental results and discussions over head pose controllability of  $\bar{w}_R$ .

**Lipschitz MLP** Since  $G$  is only trained to be conditioned over a limited set of trainable latent codes  $W$ , a subspace of  $w$  outside the learned latent codes that yields plausible deformations needs to be formulated to maximize the expressibility of  $G$  for manipulation. Meanwhile, HyperNeRF was shown to moderately render images from latent codes linearly interpolated from two learned latent codes. Thus, a valid latent subspace  $\mathcal{W}$  can be formulated to include not only the learned latent codes but codes linearly interpolated between any two learned latent codes as well. Specifically,

$$\mathcal{W} \supset \{\gamma * \bar{w}_i + (1 - \gamma) * \bar{w}_j \mid \bar{w}_i, \bar{w}_j \in \bar{W}, 0 \leq \gamma \leq 1\}. \quad (7)$$

However, we learned that the fidelity of images from

interpolated latent codes needs to be higher to be leveraged for manipulation. As so, we regularize the MLPs of the scene manipulator to be more Lipschitz continuous during its training phase. Note that Lipschitz bound of a neural network with  $L$  number of layers and piecewise linear functions such as ReLU can be approximated as  $c = \prod_{i=1}^L \|W^i\|_p$  [18, 44], where  $W^i$  is an MLP weight at  $i$ -th layer. Since a function  $f$  that is  $c$ -Lipschitz has the property

$$\|f(w_1) - f(w_2)\|_p \leq c \|w_1 - w_2\|_p, \quad (8)$$

successful regularization of  $c$  would make smaller differences between outputs of adjacent latent codes, which induce interpolated deformations to be more visually natural. As so, we follow [18] and regularize trainable matrix at  $l$ -th layer of  $F$  by introducing extra trainable parameters  $c^l$  as

$$y^l = \sigma(\hat{W}^l x + b^l), \quad \hat{W}_j^l = W_j^l \cdot \min(1, \frac{\text{softplus}(c^l)}{\|W_j^l\|_\infty}), \quad (9)$$

where  $W_j^l$  is the  $j$ -th row of a trainable matrix at  $l$ -th layer  $W^l$ , and  $\|\cdot\|_\infty$  is matrix  $\infty$ -norm. Trainable Lipschitz constants from the layers are then minimized via gradient-based optimization with loss function defined as

$$\mathcal{L}_{lip} = \prod_{l=1}^L \text{softplus}(c^l). \quad (10)$$

In summary, networks in Eq. (4) are trained to retrieve  $\bar{F}$ ,  $\bar{T}$ ,  $\bar{H}$ , and  $\bar{W}$  using our scene manipulator objective function

$$\mathcal{L}_{SM} = \lambda_c \mathcal{L}_c + \lambda_{lip} \mathcal{L}_{lip}, \quad (11)$$

where  $\lambda_c$  and  $\lambda_{lip}$  are hyper-parameters.

#### 4.2. Text-driven Manipulation

Given a trained scene manipulator  $G$ , one manipulation method is to find a single optimal latent code  $w$  whose rendered image using  $G$  yields the highest cosine similarity with a target text in CLIP[28] embedding space, so that the

manipulated images can reflect the visual attributes of a target text. Specifically, given images rendered with  $G$  and  $w$  at a set of valid camera poses  $[R|t]$  as  $\mathcal{I}_{[R|t]}^{G,w}$  and a target text for manipulation  $p$ , the goal of the method is to solve the following problem:

$$w^* = \arg \max_w D_{\text{CLIP}}(\mathcal{I}_{[R|t]}^{G,w}, p), \quad (12)$$

where  $D_{\text{CLIP}}$  measures the cosine similarity of features between rendered images and a target text extracted from pre-trained CLIP model.

As illustrated in Fig. 3b, a straightforward vanilla approach to find an optimal latent embedding  $w^*$  is inversion, a gradient-based optimization of  $w$  that maximizes Eq.(12) by defining a loss function as  $\mathcal{L}_{\text{CLIP}} = 1 - D_{\text{CLIP}}(\mathcal{I}_{[R|t]}^{G,w}, p)$ . However, we show that this method is sub-optimal by showing that it inevitably suffers from what we define as a *linked local attributes* problem, which we then solve with our proposed method.

**Linked local attribute problem** Solutions from the vanilla inversion method are confined to represent deformations equivalent to those from  $\mathcal{W}$ . However,  $\mathcal{W}$  cannot represent all possible combinations of locally observed deformations, as interpolations between two learned latent codes, which essentially comprise  $\mathcal{W}$ , cause facial attributes in different locations to change simultaneously. For example, consider a scene with deformations in Fig. 2b and renderings of interpolations between two learned latent codes in Fig. 2c. Not surprisingly, neither the learned latent codes nor the interpolated codes can express opened eyes with opened mouth or closed eyes with a closed mouth. Similar experiments can be done with any pair of learned latent codes and their interpolations to make the same conclusion.

We may approach this problem from the slicing surface perspective of canonical hyperspace introduced in Sec. 3.2. As in Fig. 2a, hyperspace allows only one latent code to represent an instance of a slicing surface representing a global deformation of all spatial locations. Such representation causes a change in one type of deformation in one location to entail the same degree of change to another type of deformation in different locations during interpolation.

Our method is motivated by the observation and is therefore designed to allow different position  $\mathbf{x}$  to be expressed with different latent codes to solve the linked local attribute problem.

### 4.3. Position-conditional Anchor Compositor

For that matter, Position-conditional Anchor Compositor (PAC) is proposed to grant our manipulation pipeline the freedom to learn appropriate latent codes for different spatial positions.

Specifically, we define anchor codes  $\{\bar{w}_1^A, \dots, \bar{w}_K^A\} = \bar{W}^A \subset \bar{W}$ , a subset of learned latent codes where each rep-

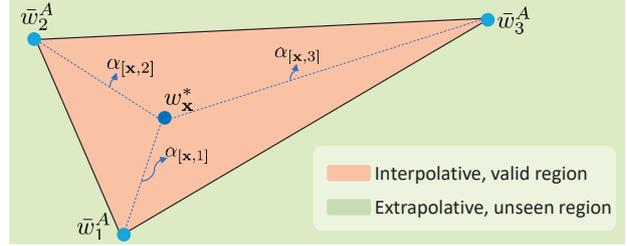


Figure 4: Illustration of barycentric interpolation of latent codes for validly expressive regions when  $K = 3$ .

resent different types of observed facial deformations, to set up a validly explorable latent space as a prior. We retrieve anchor codes by extracting facial expression parameters using DECA[5] from images rendered from all codes in  $\bar{W}$  over a fixed camera pose. Then, we cluster the extracted expression parameters using DBSCAN[3] and select the latent code corresponding to the expression parameter closest to the mean for each cluster. For instance, we may get  $K = 4$  anchor codes in the case of the example scenes in Fig. 1a and Fig. 2b.

Then for every spatial location, a position-conditional MLP yields appropriate latent codes by learning to compose these anchor codes. By doing so, a manipulated scene can be implicitly represented with multiple, point-wise latent codes. Specifically, the anchor composition network  $P : \mathbb{R}^{(3+d_w)} \rightarrow \mathbb{R}^1$  learns to yield  $w_{\mathbf{x}}^*$  for every spatial position  $\mathbf{x}$  via barycentric interpolation[8] of anchors as

$$\hat{\alpha}_{[\mathbf{x},k]} = P(\mathbf{x} \oplus \bar{w}_k^A), \quad w_{\mathbf{x}}^* = \sum_k \sigma_k(\hat{\alpha}_{[\mathbf{x},k]}) \bar{w}_k^A, \quad (13)$$

where  $d_w$  is the dimension of a latent code,  $\oplus$  is concatenation, and  $\sigma_k$  is softmax activation along  $k$  network outputs. Also, denote  $\alpha_{[\mathbf{x},k]} = \sigma_k(\hat{\alpha}_{[\mathbf{x},k]})$  as anchor composition ratio (ACR) for ease of notation. As in the illustrative example in Fig. 4, the key of the design is to prevent the composited code from diverging to extrapolative region of the latent. Thus, barycentric interpolation defines a safe bound of composited latent code for visually natural renderings.

Finally, a set of points that are sampled from rays projected at valid camera poses and their corresponding set of latent codes  $\{w_{\mathbf{x}}^*\}$  are queried by  $G$ , whose outputs are rendered as images to be supervised in CLIP embedding space for manipulation as

$$\mathcal{L}_{\text{CLIP}} = 1 - D_{\text{CLIP}}(\mathcal{I}_{[R|t]}^{G,\{w_{\mathbf{x}}^*\}}, p), \quad (14)$$

**Total variation loss on anchor composition ratio** As, the point-wise expressibility of PAC allows adjacent latent codes to vary without mutual constraints,  $P$  is regularized with total variation (TV) loss. Smoother ACR fields allows similar latent embeddings to cover certain facial po-

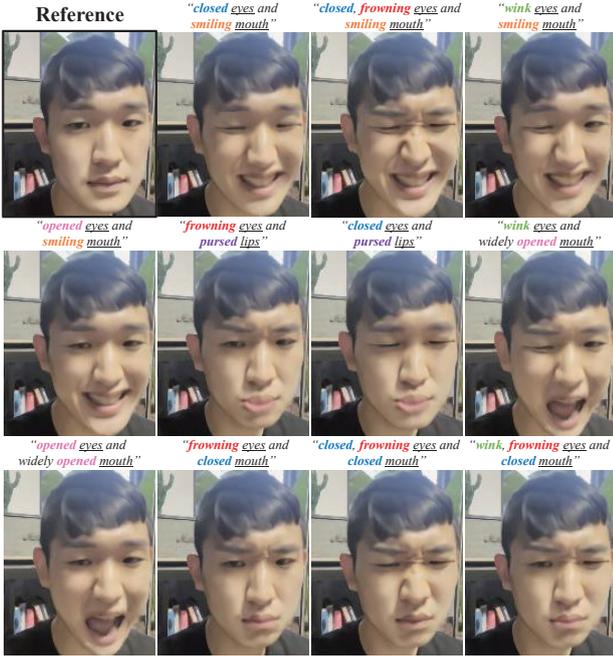


Figure 5: Qualitative results manipulated with descriptive texts using our method. Local facial deformations can easily be controlled using texts only.

sitions to yield more naturally rendered images. Specifically,  $\alpha_{[x,k]}$  is rendered to valid camera planes using the rendering equation in Eq. (1) for regularization. Given a ray  $\mathbf{r}_{uv}(t) = \mathbf{o} + t\mathbf{d}_{uv}$ , ACR can be rendered for each anchor  $k$  at an image pixel located at  $(u, v)$  of a camera plane, and regularized with TV loss as

$$\tilde{\alpha}_{kuv} = \sum_{i=1}^M T_i (1 - \exp(-\sigma_i \delta_i)) \alpha_{[\mathbf{r}_{uv}(t_i), k]}, \quad (15)$$

$$\mathcal{L}_{ACR} = \sum_{k,u,v} \|\tilde{\alpha}_{k(u+1)v} - \tilde{\alpha}_{kuv}\|_2 + \|\tilde{\alpha}_{ku(v+1)} - \tilde{\alpha}_{kuv}\|_2. \quad (16)$$

In summary, text-driven manipulation is conducted by optimizing  $P$  and minimizing the following loss

$$\mathcal{L}_{edit} = \lambda_{CLIP} \mathcal{L}_{CLIP} + \lambda_{ACR} \mathcal{L}_{ACR} \quad (17)$$

where  $\lambda_{CLIP}$  and  $\lambda_{ACR}$  are hyper-parameters.

## 5. Experiments

**Dataset** We collected portrait videos from six volunteers using Apple iPhone 13, where each volunteer was asked to make four types of facial deformations shown in Fig. 1a and Fig. 2b. A pre-trained human segmentation network was used to exclude descriptors from the dynamic part of the scenes during camera pose computation using COLMAP[32]. Examples of facial deformations observed

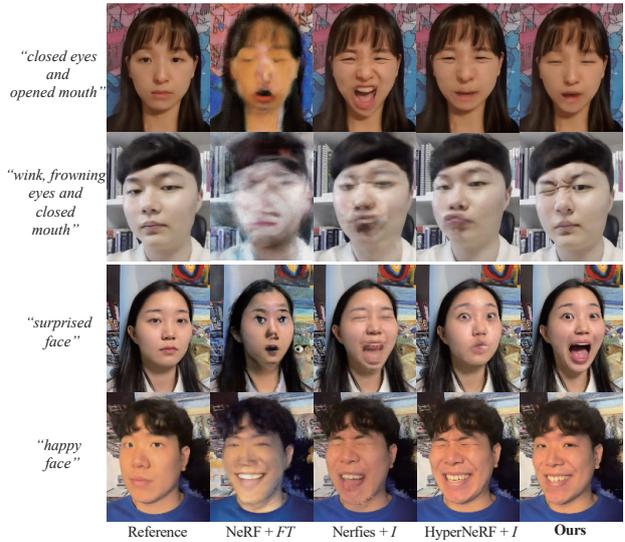


Figure 6: Text-driven manipulation results of our method and the baselines. Our result well reflects the implicit attributes of target emotional texts while preserving visual quality and face identity.

during training for each scene are reported in the supplementary material.

**Manipulation Texts** We selected two types of texts for manipulation experiments. First is a descriptive text that characterizes deformations of each facial parts. Second is an emotional expression text, which is an implicit representation of a set of multiple local deformations on *all* face parts hard to be described with descriptive texts. We selected 7 frequently used and distinguishable emotional expression texts for our experiment: "crying", "disappointed", "surprised", "happy", "angry", "scared" and "sleeping". To reduce text embedding noise, we followed [25] by averaging augmented embeddings of sentences with identical meanings.

**Implementation details**  $P$  in PAC is comprised of MLP with depth of 6 and width of 64 with ReLU activations. Optimizations for both training and manipulation were conducted using Adam[14] on a single NVIDIA A100 GPU. Gradient calculations per image require computations from all sampled points over rays projected from all pixels, which causes GPU memory issue. As so, we set the training frame resolution to 240x135, and computed gradients to random portions of rays per iteration following [49]. constructing several sentences with identical meanings and use the average of their CLIP embeddings.

**Baselines** Since there is no prior work that is parallel to our problem definition, we formulated 3 baselines with existing state-of-the-art methods for comparisons: (1) NeRF+FT is a simple extension from NeRF [21] that fine-tunes

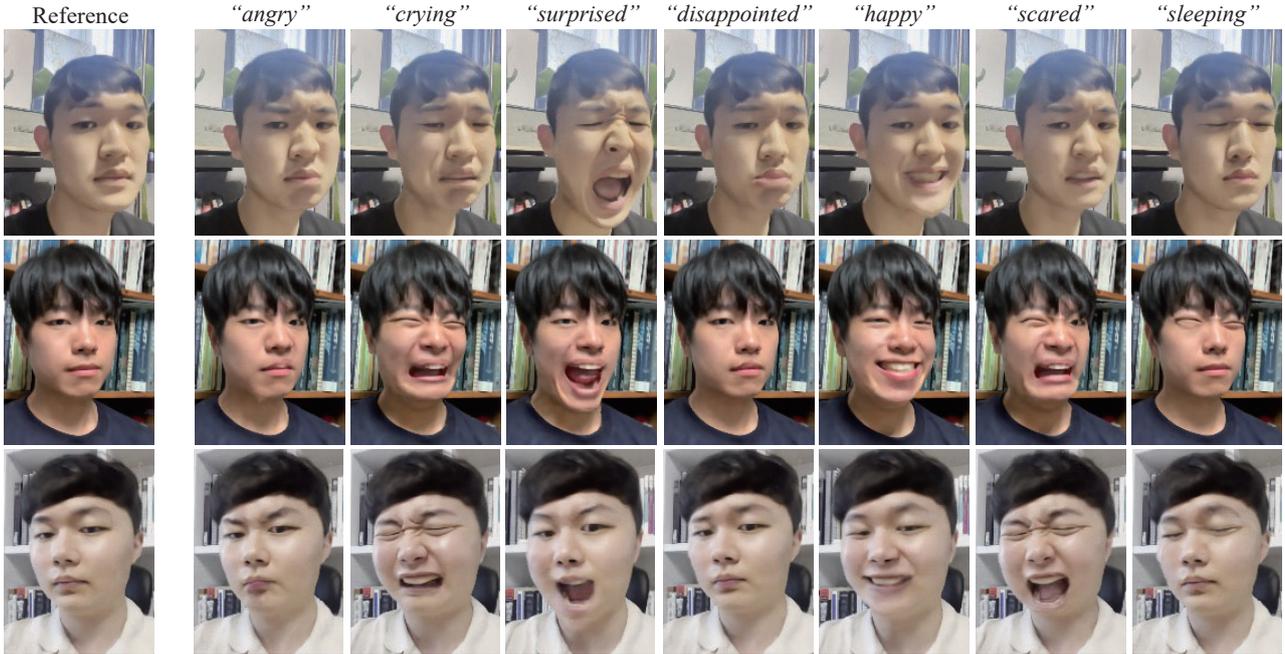


Figure 7: Extensive face manipulation results driven by a set of frequently used emotional expression texts using our method. Manipulating to emotional expression texts are challenging, as they implicitly require compositions of subtle facial deformations that are hard to be described. Our method reasonably reflects the attributes of the manipulation texts.

the whole network using CLIP loss, (2) Nerfies+ $I$  uses Nerfies[23] as a deformation network followed by conducting vanilla inversion method introduced in Sec. §4.2 for manipulation, and (3) HyperNeRF+ $I$  replaces Nerfies in (2) with HyperNeRF [24].

**Text-driven Manipulation** We report qualitative manipulation results of our methods driven with a set of descriptive sentences in Fig. 5. Our method not only faithfully reflects the descriptions, but also can easily control local facial deformations with simple change of words in sentences. We also report manipulated results driven by emotional expression texts in Fig. 7. As can be seen, our method conducts successful manipulations even if the emotional texts are implicit representations of many local facial deformations. For instance, result manipulated with ”crying” in first row of Fig. 7 is not expressed with mere crying-looking eyes and mouth, but also includes crying-looking eyebrows and skin all over the face without any explicit supervision on local deformations. We also compare our qualitative results to those from the baselines in Fig. 6. Ours result in the highest reflections of the target text attributes. NeRF+ $FT$  shows significant degradation in visual quality, while Nerfies+ $I$  moderately suffers from low reconstruction quality and reflection of target text attributes. HyperNeRF+ $I$  shows the highest visual quality out of all baselines, yet fails to reflect the visual attributes of target texts.

High reflectivity on various manipulation texts can be at-

	R-Prec.[41] $\uparrow$	LPIPS[47] $\downarrow$	CFS $\uparrow$
NeRF + $FT$	<u>0.763</u>	0.350	0.350
Nerfies + $I$	0.213	0.222	0.684
HyperNeRF + $I$	0.342	<u>0.198</u>	<u>0.721</u>
Ours	<b>0.780</b> (+0.017)	<b>0.082</b> (-0.116)	<b>0.749</b> (+0.028)

Table 1: Quantitative results. R-Prec. denotes R-precision, and CFS denotes cosine face similarity. We notate performance ranks as **best** and second best.

	TR $\uparrow$	VR $\uparrow$	FP $\uparrow$
NeRF + $FT$	<u>2.85</u>	0.18	0.79
Nerfies + $I$	0.33	3.61	4.03
HyperNeRF + $I$	2.52	<u>4.42</u>	<u>4.39</u>
Ours	<b>4.15</b> (+1.30)	<b>4.58</b> (+0.16)	<b>4.67</b> (+0.28)

Table 2: User study results. TR, VR, and FP denote text reflectivity, visual realism, and face identity preservability, respectively. **Best** and second best are highlighted.

tributed to PAC that resolves the linked local attribute problem. In Fig. 8, we visualize  $\bar{\alpha}_{kuv}$  for each anchor code  $k$ , which is the rendering of ACR  $\alpha_{[x,k]}$  in Eq. (15), over an image plane. Whiter regions of the renderings are closer to one, which indicates that the corresponding anchor code is mostly composited to yield the latent code of the region. Also, we display image renderings from each anchor code

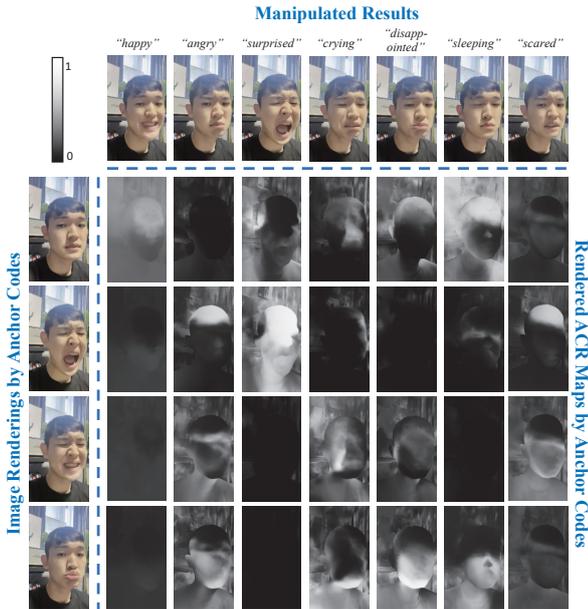


Figure 8: Renderings of learned ACR maps for each anchor codes over different manipulation texts.

on the left to help understand the local attributes for each anchor code. As can be seen, PAC composes appropriate anchor codes for different positions. For example, when manipulating for *sleeping* face, PAC reflects closed eyes from one anchor code and neutral mouth from other anchor codes. In the cases of *crying*, *angry*, *scared*, and *disappointed* face, PAC learns to produce complicated compositions of learned deformations, which are inexpressible with a single latent code.

**Quantitative Results** First of all, we measured R-precision[41] to measure the text attribute reflectivity of the manipulations. We used facial expression recognition model[31] pre-trained with AffectNet[22] for top-R retrievals of each text. Specifically, 1000 novel view images are rendered per face, where 200 images are rendered from a face manipulated with each of the five texts that are distinguishable and exist in AffectNet labels: *“happy”*, *“surprised”*, *“fearful”*, *“angry”*, and *“sad”*. Also, to estimate the visual quality after manipulation, we measured LPIPS[47] between faces with no facial expressions (neutral faces) without any manipulations and faces manipulated with 7 texts, each of which are rendered from 200 novel views. Note that LPIPS was our best estimate of visual quality since there can be no pixel-wise ground truth of text-driven manipulations. Lastly, to measure how much of the facial identity is preserved after manipulation, we measured the cosine similarity between face identity features<sup>1</sup> extracted from neutral faces and text-manipulated faces, all

<sup>1</sup><https://github.com/ronghuaiyang/arcface-pytorch>



Figure 9: Renderings from linearly interpolated latent codes. Lipschitz-regularized scene manipulator interpolates unseen shapes more naturally.

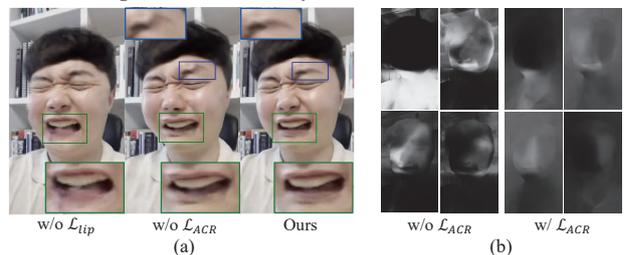


Figure 10: (a) Qualitative results of the ablation study. Manipulations are done using *“crying face”* as target text. (b) Rendered ACR maps with and without  $\mathcal{L}_{ACR}$ .

of which are rendered from 200 novel views. Table 1 reports the average results over all texts, which shows that our method outperforms in all criteria.

**User Study** Users were asked to score from 0 to 5 on 3 criteria; (i) Text Reflectivity: how well the manipulated renderings reflect the target texts, (ii) Visual Realism: how realistic do the manipulated images look, and (iii) Face identity Preservability: how well do the manipulated images preserve the identity of the original face, over our method and the baselines. The following results are reported in Table. 2. Our method outperforms all baselines, and especially in text reflectivity by a large margin. Note that the out-performance in user responses align with that from the quantitative results, which supports the consistency of evaluations.

**Interpolation** We experiment with the effect of Lipschitz regularization on the scene manipulator by comparing the visual quality of images rendered from linearly interpolated latent codes, and report the results in Fig. 9. Lipschitz-regularized scene manipulator yields more visually natural images, which implies that learned set of anchor-composited latent codes  $[w_x^*]$  are more likely to render realistically interpolated local deformations under Lipschitz-regularized scene manipulator.

**Ablation Study** We conducted an ablation study on our regularization methods:  $\mathcal{L}_{lip}$  and  $\mathcal{L}_{ACR}$ . As shown in Fig. 10a, manipulation without  $\mathcal{L}_{lip}$  suffers from low visual quality. Manipulation without  $\mathcal{L}_{ACR}$  yields unnatural renderings of face parts with large deformation range such as mouth and eyebrows. This can be interpreted with learned ACR maps of PAC in Fig. 10b. ACR maps learned with  $\mathcal{L}_{ACR}$  introduces reasonable continuities of latent codes on boundaries of the dynamic face parts, thus yielding naturally interpolated face parts.

## 6. Conclusion

We have presented FaceCLIPNeRF, a text-driven manipulation pipeline of a 3D face using deformable NeRF. We first proposed a Lipschitz-regularized scene manipulator, a conditional MLP that uses its latent code as a control handle of facial deformations. We addressed the linked local attribute problem of conventional deformable NeRFs, which cannot compose deformations observed in different instances. As so, we proposed PAC that learns to produce spatially-varying latent codes, whose renderings with the scene manipulator were trained to yield high cosine similarity with target text in CLIP embedding space. Our experiments showed that our method could faithfully reflect the visual attributes of both descriptive and emotional texts while preserving visual quality and identity of 3D face.

**Acknowledgement** This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-22-1-4024, KAIST-NAVER hypercreative AI center, and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).

## References

- [1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.
- [2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.
- [3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- [4] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. *arXiv preprint arXiv:2204.01943*, 2022.
- [5] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. volume 40, 2021.
- [6] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2021.
- [7] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20374–20384, 2022.
- [8] Kai Hormann. Barycentric interpolation. In *Approximation Theory XIV: San Antonio 2013*, pages 197–218. Springer, 2014.
- [9] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022.
- [10] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. 2022.
- [11] Nikolay Jetchev. Clipmatrix: Text-controlled creation of 3d textured meshes. *arXiv preprint arXiv:2109.12922*, 2021.
- [12] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.
- [13] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, and Andrea Tagliasacchi. CoNeRF: Controllable Neural Radiance Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [15] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- [16] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. *arXiv preprint arXiv:2204.10850*, 2022.
- [17] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022.
- [18] Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. *arXiv preprint arXiv:2202.08345*, 2022.
- [19] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5773–5783, 2021.

- [20] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022.
- [21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [22] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017.
- [23] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.
- [24] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.
- [25] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *International Conference of Computer Vision*, pages 2085–2094, 2021.
- [26] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [27] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020.
- [28] 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. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
- [29] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629, 2021.
- [30] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshah. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, 2022.
- [31] Andrey V Savchenko. Frame-level prediction of facial expressions, valence, arousal and action units for mobile devices. *arXiv preprint arXiv:2203.13436*, 2022.
- [32] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [33] Sahil Sharma and Vijay Kumar. 3d face reconstruction in deep learning era: A survey. *Archives of Computational Methods in Engineering*, pages 1–33, 2022.
- [34] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *arXiv preprint arXiv:2205.15517*, 2022.
- [35] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022.
- [36] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021.
- [37] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2387–2395, 2016.
- [38] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022.
- [39] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022.
- [40] Ziyang Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. Learning compositional radiance fields of dynamic human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5704–5713, 2021.
- [41] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.
- [42] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *International Conference on Computer Vision (ICCV)*, October 2021.
- [43] Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. i3dmm: Deep implicit 3d morphable model of human heads. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12803–12813, 2021.
- [44] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

- [45] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.
- [46] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022.
- [47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [48] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13545–13555, 2022.
- [49] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. 2021.
- [50] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofan-erf: Morphable facial neural radiance field. *arXiv preprint arXiv:2112.02308*, 2021.