

# MoFaNeRF: Morphable Facial Neural Radiance Field

Yiyu Zhuang\* Hao Zhu\* Xusen Sun Xun Cao  
Nanjing University, Nanjing, China

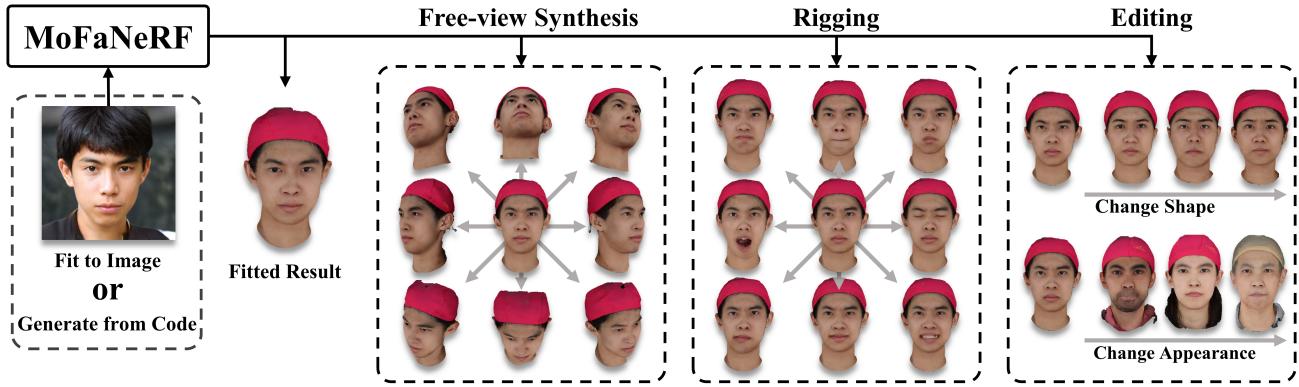


Figure 1. MoFaNeRF is a parametric model that can synthesize free-view images by fitting to a single image or generating from a random code. The synthesized face is *morphable* that can be rigged to a certain expression and be edited to a certain shape or appearance.

## Abstract

We propose a parametric model that maps free-view images into a vector space of coded facial shape, expression and appearance using a neural radiance field, namely Morphable Facial NeRF. Specifically, MoFaNeRF takes the coded facial shape, expression and appearance along with space coordinate and view direction as input to an MLP, and outputs the radiance of the space point for photo-realistic image synthesis. Compared with conventional 3D morphable models (3DMM), MoFaNeRF shows superiority in directly synthesizing photo-realistic facial details even for eyes, mouths, and beards. Also, continuous face morphing can be easily achieved by interpolating the input shape, expression and appearance codes. By introducing identity-specific modulation and texture encoder, our model synthesizes accurate photometric details and shows strong representation ability. Our model shows strong ability on multiple applications including image-based fitting, random generation, face rigging, face editing, and novel view synthesis. Experiments show that our method achieves higher representation ability than previous parametric models, and achieves competitive performance in several applications. To the best of our knowledge, our work is the first facial parametric model built upon a neural radiance field that can be used in fitting, generation and manipulation. Our code and model are released in <https://github.com/zhuhao-nju/mofanerf>.

\* These authors contributed equally to this work.

## 1. Introduction

Modeling 3D face is a key problem to solve face-related vision tasks such as 3D face reconstruction, reenactment, parsing, and digital human. The 3D morphable model (3DMM) [3] has long been the key solution to this problem, which is a parametric model transforming the shape and texture of the faces into a vector space representation. 3DMMs are powerful in representing various shapes and appearances, but require a sophisticated rendering pipeline to produce photo-realistic images. Besides, 3DMMs struggled to model non-Lambertian objects like pupils and beards. Recently, the neural radiance field (NeRF) [31] was proposed to represent the shapes and appearances of a static scene using an implicit function, which shows superiority in the task of photo-realistic free-view synthesis. The most recent progress shows that the modified NeRF can model a dynamic face [11, 33, 34, 51], or generate diversified 3D-aware images [5, 15, 40]. However, there is still no method to enable NeRF with the abilities of single-view fitting, controllable generation, face rigging and editing at the same time. In summary, conventional 3DMMs are powerful in representing large-scale editable 3D faces but lack the ability of photo-realistic rendering, while NeRFs are the opposite.

To combine the best of 3DMM and NeRF, we aim at creating a facial parametric model based on the neural radiance field to have the powerful representation ability as well as excellent free-view rendering performance. However, achieving such a goal is non-trivial. The challenges

come from two aspects: firstly, how to memorize and parse the very large-scale face database using a neural radiance field; secondly, how to effectively disentangle the parameters (e.g. shape, appearance, expression), which is important to support very valuable applications like face rigging and editing.

To address these challenges, we propose the Morphable Facial NeRF (MoFaNeRF) that maps free-view images into a vector space of coded facial identity, expression, and appearance using a neural radiance field. Our model is trained on 720,000 multi-view images from FaceScape [53, 59] dataset, containing over 6000 faces of 359 different subjects with 20 expressions each. The training strategy is elaborately designed to disentangle the shape, appearance and expression in the parametric space. The identity-specific modulation and texture encoder are proposed to maximize the representation ability of the neural network. Compared to traditional 3DMMs, MoFaNeRF shows superiority in synthesizing photo-realistic images even for pupils, mouth, and beards which can not be modeled well by 3D mesh models. Furthermore, we also propose the methods to use our model to achieve image-based fitting, random face generation, face rigging, face editing, and view extrapolation. Our contributions can be summarized as follows:

- To the best of our knowledge, we propose the first parametric model that maps free-view images into a vector space using a neural radiance field, which supports both controllable generation and image-based fitting.
- The network and the training strategy are elaborately designed to maximize the representation ability of the model. The disentangled parameters of shape, appearance and expression can be interpolated to achieve smoothly morphable synthesis.
- We propose the methods to use our model for multiple applications including image-based fitting, view extrapolation, face editing and face rigging, and achieve competitive performance compared to state-of-the-art methods.

## 2. Related Work

As our work is a parametric model based on neural radiance field, we will review the related work of 3D morphable model and neural radiance field respectively.

**3D Morphable Model.** 3DMM is a statistical model which transforms the shape and texture of the faces into a vector space representation [3]. By optimizing and editing parameters, 3DMMs can be used in multiple applications like 3D face reconstruction, alignment, animation, etc. We recommend referring to the recent survey [9] for a comprehensive review of 3DMM. To build a 3DMM, traditional approaches first capture a large number of 3D facial meshes, then align them into a uniform topology representation, and

finally process them with principal component analysis algorithm [4, 18, 27, 48, 53]. The parameter of the 3DMM can be further disengaged into multiple dimensions like identity, expression, appearance, and poses. In recent years, several works tried to enhance the representation power of 3DMM by using a non-linear mapping [2, 7, 42, 44–46], which is more powerful in representing detailed shape and appearance than transitional linear mapping. However, they still suffer from the mesh representation which is hard to model fine geometry of pupils, eyelashes and hairs. Besides, traditional 3DMMs require sophisticated rendering pipelines to render photo-realistic images. By contrast, our model doesn't explicitly generate shape but directly synthesizes photo-realistic free-view images even for pupils, inner-mouth and beards.

Very recently, Yenamandra *et al.* [54] proposed to build the 3DMM with an implicit function representing facial shape and appearance. They used a neural network to learn a signed distance field(SDF) of 64 faces, which can model the whole head with hair. Similarly, our model is also formulated as an implicit function but very different from SDF. SDF still models shape while our method focuses on view synthesis and releases constraints of the shape, outperforming SDF in rendering performance by a large margin.

**Neural Radiance Field.** NeRF [31] was proposed to model the object or scene with an impressive performance in free-view synthesis. NeRF synthesizes novel views by optimizing an underlying continuous volumetric scene function that is learned from multi-view images.

As the original NeRF is designed only for a static scene, many efforts have been devoted to reconstructing deformable objects. Aiming at the human face many methods [11, 34, 51] modeled the motion of a single human head with a designed conditional neural radiance field, extending NeRF to handle dynamic scenes from monocular or multi-view videos. Aiming at human body, several methods have been proposed by introducing human parametric model (e.g. SMPL) [6, 28, 32, 36] or skeleton [35] as prior to build NeRF for human body. For a wide range of dynamic scenarios, Park *et al.* [33] proposed to augment NeRF by optimizing an additional continuous volumetric deformation field, while Pumarola *et al.* [37] optimized an underlying deformable volumetric function. Another group of works [5, 15, 40] turned NeRF into a generative model that is trained or conditioned on certain priors, which achieves 3D-aware images synthesis from a collection of unposed 2D images. To reduce the image amount for training, many works [12, 39, 49, 55] trained the model across multiple scenes to learn a scene prior, which achieved reasonable novel view synthesis from a sparse set of views.

Different from previous NeRFs, our method is the first parametric model for facial neural radiance field trained on a large-scale multi-view face dataset. Our model sup-

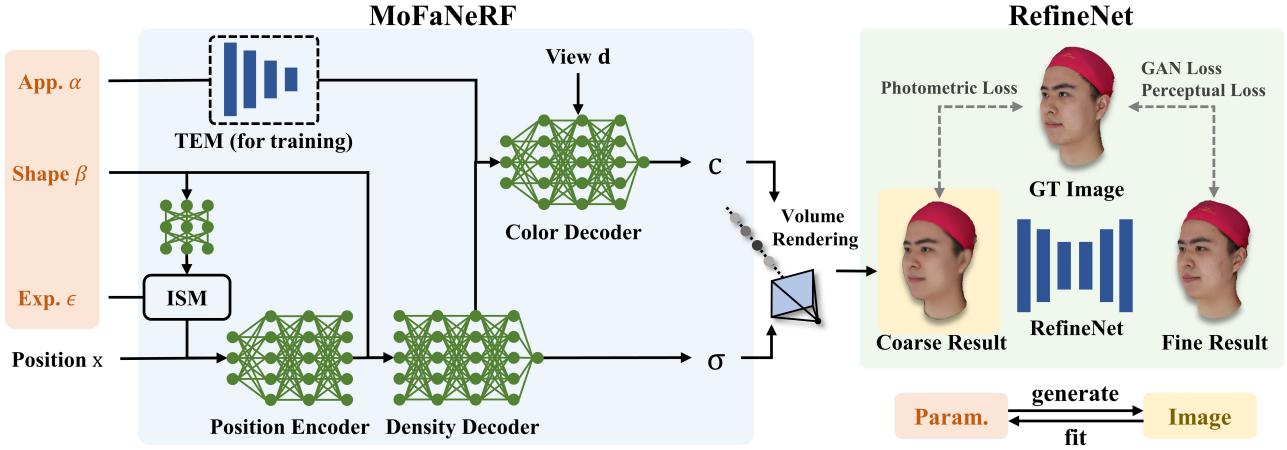


Figure 2. MoFaNeRF takes appearance code  $\alpha$ , shape code  $\beta$ , expression code  $\epsilon$ , position code  $x$  and view direction  $d$  as input, synthesizing a coarse result which is then refined by a RefineNet. As shown in the right bottom corner, MoFaNeRF can be used in generating (synthesize free-view images given parameters) or fitting (optimize for parameters given a single image).

ports multiple applications including random face generation, image-based fitting and facial editing, which is unavailable for previous NeRFs.

### 3. Morphable Facial NeRF

Morphable facial NeRF is a parametric model that maps free-view facial portraits into a continuous morphable parametric space, which is formulated as:

$$\mathcal{M} : (\mathbf{x}, \mathbf{d}, \beta, \alpha, \epsilon) \rightarrow \{\mathbf{c}, \sigma\}, \quad (1)$$

where  $\mathbf{x}$  is the 3D position of a sample point;  $\mathbf{d}$  is the viewing direction consisting of pitch and yaw angles;  $\beta, \alpha, \epsilon$  are the parameters denoting facial shape, appearance, and expression respectively;  $\mathbf{c}$  and  $\sigma$  are the RGB color and the density used to represent the neural radiance field. In the next, we will explain  $\mathbf{x}, \mathbf{d}, \mathbf{c}, \sigma$  that are referred from NeRF in Section 3.1, then introduce  $\beta, \alpha, \epsilon$  in Section 3.2. The network design is illustrated in Section 3.3 and the training details are explained in Section 3.4.

#### 3.1. Neural Radiance Field

As defined in NeRF [31], the radiance field is represented as volumetric density  $\sigma$  and color  $\mathbf{c} = (R, G, B)$ . An MLP is used to predict  $\sigma$  and  $\mathbf{c}$  from a 3D point  $\mathbf{x} = (x, y, z)$  and viewing direction  $\mathbf{d} = (\theta, \phi)$ . Position encoding is introduced to transform the continuous inputs  $\mathbf{x}$  and  $\mathbf{d}$  into a high-dimensional space, which is also used in our model. The field of  $\sigma$  and  $\mathbf{c}$  can be rendered to images using a differentiable volume rendering module. For a pixel in the posed image, a ray  $\mathbf{r}$  is cast through the neural volume field from the ray origin  $\mathbf{o}$  along the ray direction  $\mathbf{d}$  according to the camera parameters, which is formulated as  $\mathbf{r}(z) = \mathbf{o} + zd$ . Through sampling points along this ray,

and accumulating the sampled density  $\sigma(\cdot)$  and RGB values  $\mathbf{c}(\cdot)$  computed by  $\mathcal{F}$ , the final output color  $\mathbf{C}(\mathbf{r})$  of this pixel can be evaluated by:

$$\mathbf{C}(\mathbf{r}) = \int_{z_n}^{z_f} T(z)\sigma(\mathbf{r}(z))\mathbf{c}(\mathbf{r}(z), \mathbf{d})dz, \quad (2)$$

where  $z_n$  and  $z_f$  are near and far bounds.  $T(t)$  is defined as the accumulated transmittance along the ray from  $z_n$  to  $z$ :

$$T(z) = \exp\left(-\int_{z_n}^z \sigma(\mathbf{r}(s))ds\right). \quad (3)$$

Through the rendered color, a photometric loss can be applied to supervise the training of the MLP.

#### 3.2. Parametric Mapping

Our model is conditioned on the parameters to represent the identity and facial expression  $\epsilon$ , and the identity is further divided into shape  $\beta$  and appearance  $\alpha$ . Initially, we consider integrating  $\beta$  and  $\alpha$  into a single identity code, however, we find it is hard for an MLP to memorize the huge amount of appearance information. Therefore, we propose to decouple the identity into shape and appearance. These parameters need to be disentangled to support valuable applications like face rigging and editing.

**Shape parameter**  $\beta$  represents the 3D shape of the face that is only related to the identity of the subject, like the geometry and position of the nose, eyes, mouth and overall face. A straightforward idea is to use one-hot encoding to parameterize  $\beta$ , while we find it suffers from redundant parameters because the similarity of large-amount faces is repeatedly expressed in one-hot code. Instead, we adopt the identity parameters of the bilinear model of FaceScape [53] as shape parameter, which is the PCA factors of the 3D mesh for each

subject. The numerical variation of the identity parameter reflects the similarity between face shapes, which makes the solution space of facial shapes more efficient.

**Appearance parameter**  $\alpha$  reflects photometric features like the colors of skin, lips, and pupils. Some fine-grained features are also reflected by appearance parameters, such as beard and eyelashes. Considering that the UV texture provided by FaceScape dataset is the ideal carrier to convey the appearance in a spatial-aligned UV space, we propose to encode the UV texture maps into  $\alpha$  for training. The texture encoding module (TEM) is proposed to transfer the coded appearance information into the MLP, which is a CNN based encoder network. TEM is only used in the training phase, and we find it significantly improves the quality of synthesized images. We consider the reason is that the appearance details are well disentangled from shape and spatial-aligned, which relieves the burden of memorizing appearances for the MLP.

**Expression parameter**  $\epsilon$  is corresponding to the motions caused by facial expressions. Previous methods [34, 47] try to model the dynamic face by adding a warping vector to the position code  $\mathbf{x}$ , namely deformable volume. However, our experiments show that the deformable volume doesn't work in our task where too many subjects are involved in a single model. More importantly, our training data are not videos but images with discrete 20 expressions, which makes it even harder to learn a continuous warping field. By contrast, we find directly concatenating expression parameters with the position code causes fewer artifacts, and our identity-specific modulation (detailed in Section 3.3) further enhances the representation ability of expression. We are surprised to find that MLP without a warping module can still synthesize continuous and plausible interpolation for large-scale motions. We believe this is the inherent advantage of the neural radiance field over 2D-based synthesis methods.

### 3.3. Network Design

As shown in Figure 2, the backbone of MoFaNeRF mainly consists of position encoder, color decoder, density decoder and identity-specific modulation. These networks transform the parameters  $\alpha, \beta, \epsilon$ , position code  $\mathbf{x}$  and viewing direction  $\mathbf{d}$  into the color  $\mathbf{c}$  and density  $\sigma$ . The predicted colors are then synthesized from  $\mathbf{x}$  and  $\sigma$  through volume rendering. Considering that the appearance code  $\alpha$  is only related to the color  $c$ , it is only fed into the color decoder. The expression code  $\epsilon$  is concatenated to the position code after the identity-specific modulation, as it mainly reflects the motions that are intuitively modulated by shape  $\beta$ . The additional texture encoder module (TEM) is used only in the training phase. The RefineNet takes the coarse image predicted by MoFaNeRF as input and synthesizes a refined face. The results presented in this paper are the refined re-

sults by default. Detailed parameters of our network can be found in the appendix.

We try to improve the capacity of the network by increasing the number of layers in MLP and the number of nodes in hidden layers. The prediction indeed gets improved after enlarging the model size, but is still blurry and contains artifacts when the range of expression is large. Besides, a larger model leads to slower convergence and higher memory consumption. To further improve the performance, we propose to the identity-specific modulation and RefineNet. **Identity-specific modulation (ISM).** Intuitively, facial expressions of different individuals differ from each other as individuals have their unique expression idiosyncrasies. However, we observed that the MLPs erase most of these unique characteristics after the disentanglement, homogenizing the expressions from different subjects. Motivated by AdaIN [20, 21], we consider the unique expression of individuals as a modulation relationship between  $\beta$  and  $\epsilon$ , which can be formulated as:

$$\epsilon' = M_s(\beta) \cdot \epsilon + M_b(\beta), \quad (4)$$

where  $\epsilon'$  is the updated value to the expression code,  $M_s$  and  $M_b$  are the shallow MLPs to transform  $\beta$  into an identity-specific code to adjust  $\epsilon$ . Both  $M_s$  and  $M_b$  output tensors with the same length as  $\epsilon$ . Our experiments show that ISM improves the representation ability of the network especially for various expressions.

**RefineNet.** It is hard for a single MLP-based neural network to memorize the large-scale facial details, therefore, we use a RefineNet to synthesize details based on the prediction of MoFaNeRF. We use Pix2PixHD [50] as the backbone of MoFaNeRF, which fully takes advantage of the GAN loss and perceptual loss. The input of RefineNet is the coarse image rendered by MoFaNeRF, and the output is a refined image with high-frequency details.

### 3.4. Training

**Data preparation.** We use 7180 multi-view models released by FaceScape to train our model. The models are captured from 359 different subjects with 20 expressions each. We randomly select 300 subjects (6000 scans) as training data, leaving 59 subjects (1180 scans) for testing. All these models are aligned in a canonical space, and the area below the shoulder is removed. Roughly 860,000 images are rendered in 120 viewpoints with 720,000 as training set and the rest as testing set. The details about the rendering can be found in the appendix.

**Landmark-based sampling.** In the training phase, the frequency of ray-sampling is modified besides the uniform sampling to make the network focus on the facial region. Specifically, we get 64 2D landmark key-points of the mouth, nose, eyes, and eyebrows provided by FaceScape, and the inverse-projecting rays are sampled around each



Figure 3. Random generated results by our model.

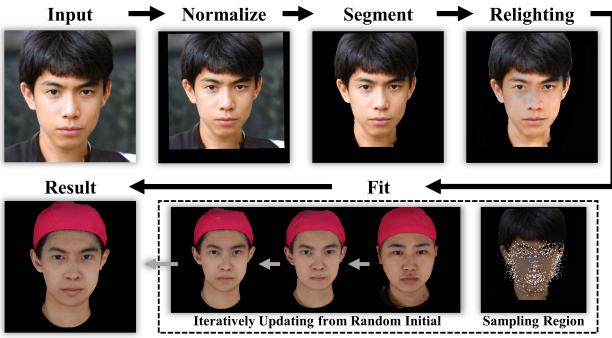


Figure 4. The pipeline for fitting our model to a single image.

key-point based on a Gaussian distribution with standard deviation equals to 0.025 of the image size. The uniform sampling and the landmark-based sampling are combined with the ratio of 2:3.

**Loss function.** The loss function to train MoFaNeRF is formulated as:

$$L = \sum_{\mathbf{r} \in \mathcal{R}} \left[ \left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 + \left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2 \right], \quad (5)$$

where  $\mathcal{R}$  is the set of rays in each batch,  $C(\mathbf{r})$  is the ground-truth color,  $\hat{C}_c(\mathbf{r})$  and  $\hat{C}_f(\mathbf{r})$  are the colors predicted by coarse volume and fine volume along ray  $\mathbf{r}$  respectively. It is worth noting that the expression and appearance parameters are updated according to the back-propagated gradient in the training, while the shape parameters remain unchanged. We firstly train the network of MoFaNeRF, then keep the model fixed and train the RefineNet. The RefineNet is trained with the loss function following Pix2PixHD [50], which is the combination of GAN loss [14] and perceptual loss [8, 13, 19]. The implementation details can be found in the appendix.

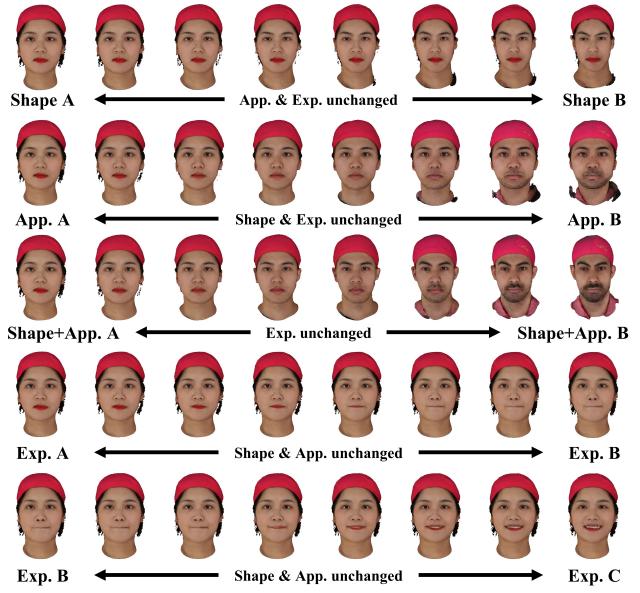


Figure 5. The generated faces can morph smoothly by interpolating in the vector space. We are surprised to find the plausible morphing results between large-scale motions (like mouth opening), though no warping module is applied.

## 4. Application

### 4.1. Free-view Generation

Once trained, MoFaNeRF can synthesize free-view human faces given a certain or random vector. We verify that the generated face can morph smoothly by interpolating the code of shape, appearance and expression, as shown in Figure 5. For the random generation, Gaussian distribution is applied to  $\beta$ ,  $\alpha$ , and  $\epsilon$  to constrain the distribution within a reasonable space. The randomly generated results are shown in Figure 3, and more can be found in the appendix.

### 4.2. Image-based Fitting

As shown in Figure 4. We propose to fit our model to an input image. Firstly, we normalize the image to the canonical image with an affine transformation. Specifically, we first extract the 2D landmarks  $L_t$  from the target image with [22], then align  $L_t$  to the predefined 3D landmarks  $L_c$  of the canonical face by solving:

$$\mathbf{d}, \mathbf{s} = \arg \min \| (\Pi(L_c, \mathbf{d})) \cdot \mathbf{s} - L_t \|_2, \quad (6)$$

where  $\mathbf{d}$  is the view direction,  $\mathbf{s}$  is the scale.  $\Pi(L_c, \mathbf{d})$  is the function to project 3D points to the 2D image plane according to the view direction  $\mathbf{d}$ . The scale  $\mathbf{s}$  is applied to the target image, and  $\mathbf{d}$  is used in the fitting and remains constant. Then we use EHANet [25, 29] to segment the background out, and normalize the lighting with the relighting method [57]. In practice, we find it important to eliminate

Table 1. Quantitative evaluation of representation ability.

Model	PSNR(dB) $\uparrow$	SSIM* $\uparrow$	LPIPS* $\downarrow$
FaceScape [53]	27.96 $\pm$ 1.34	9.32 $\pm$ 0.12	0.69 $\pm$ 0.09
FaceScape-e [53]	27.07 $\pm$ 1.46	9.33 $\pm$ 0.11	0.80 $\pm$ 0.14
i3DMM [54]	24.45 $\pm$ 1.58	9.04 $\pm$ 0.14	1.12 $\pm$ 0.15
MoFaNeRF	<b>31.49<math>\pm</math>1.75</b>	<b>9.51<math>\pm</math>0.10</b>	0.61 $\pm$ 0.11
MoFaNeRF-fine	30.17 $\pm$ 1.71	9.35 $\pm$ 0.13	<b>0.34<math>\pm</math>0.07</b>

\*The values of SSIM and LPIPS are multiplied by 10.

the influence of light because our model cannot model complex lighting well.

After the pre-processing, we can optimize for  $\beta, \alpha, \epsilon$  through the network. Specifically,  $\beta$  and  $\alpha$  are randomly initialized as explained in Section 5.4, and  $\epsilon$  is initialized with the learned value from the training. Then we freeze the pre-trained network weights and optimize  $\alpha, \beta, \epsilon$  through the network by minimizing only the MSE loss function between the predicted color and the target color. Only points around landmarks are sampled in fitting.

### 4.3. Face Rigging and Editing

The generated or fitted face can be rigged by interpolating in expression dimension with controllable view-point. The expression vector can be obtained by fitting to a video or manually set. Currently, we only use the basic 20 expressions provided by FaceScape to generate simple expression-changing animation. By improving the rigging of the face to higher dimensions [26], our model has the potential to perform more complex expressions. The rigged results are shown in Section 1 and the appendix.

The generated or fitted face can be edited by manipulating the shape and appearance code. As explained in Section 3.2, shape coder refers to the shape of the face, the geometry and position of the nose, eyes, and mouth; while appearance refers to the color of skin, lips, pupils, and fine-grained features like beard and eyelashes. These features can be replaced from face A to face B by simply replacing the shape or appearance code, as shown in Figure 1. Our model supports manually editing by painting texture map, then using TEM to generate appearance code for a generation. However, we find only large-scale features of the edited content in the texture map will take effect, like skin color and beard, while small-scale features like moles won't be transferred to the synthesized face. We also demonstrate that the face can morph smoothly by interpolating in the vector space, as shown in Figure 5.

## 5. Experiment

We firstly compare our model with previous works in representation ability, then show the effectiveness of the parameter disentanglement and the network design in the abla-

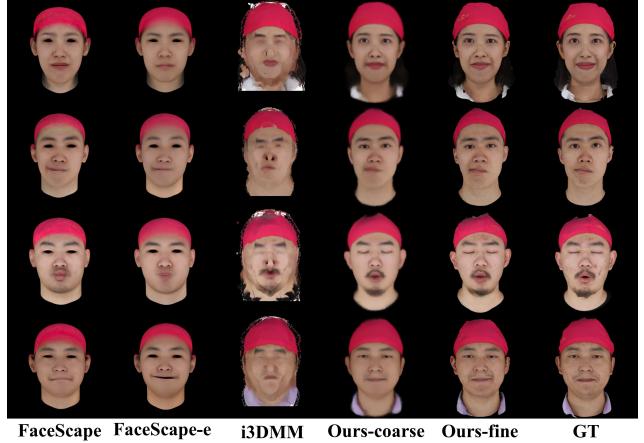


Figure 6. Visual comparison of representation ability.

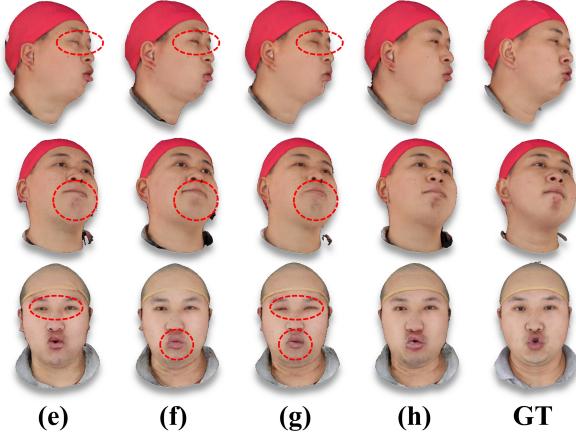


Figure 7. Visual comparison of ablation study.

Table 2. Quantitative evaluation of ablation study.

Label	PSNR(dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
(a)	25.59 $\pm$ 2.25	0.888 $\pm$ 0.025	0.184 $\pm$ 0.039
(b)	25.79 $\pm$ 2.25	0.886 $\pm$ 0.025	0.187 $\pm$ 0.039
(c)	26.27 $\pm$ 2.25	0.895 $\pm$ 0.024	0.174 $\pm$ 0.039
(d)	25.24 $\pm$ 2.13	0.883 $\pm$ 0.024	0.200 $\pm$ 0.041
(e)	25.73 $\pm$ 2.03	0.889 $\pm$ 0.025	0.184 $\pm$ 0.039
(f)	25.69 $\pm$ 2.22	0.886 $\pm$ 0.025	0.187 $\pm$ 0.039
(g)	25.67 $\pm$ 2.09	0.888 $\pm$ 0.024	0.185 $\pm$ 0.037
(h)	<b>26.57<math>\pm</math>2.08</b>	<b>0.897<math>\pm</math>0.025</b>	<b>0.166<math>\pm</math>0.037</b>

tion study. Finally, we evaluate the performance of MoFaNeRF in random generating, single-view image-based fitting, view extrapolation, and face manipulation.

### 5.1. Comparison of Representation Ability

We compare the representation ability of our MoFaNeRF with two SOTA facial parametric models - FaceScape bilinear model [53] and i3DMM [54]. FaceScape is the traditional 3DMM that applies PCA to 3D triangle mesh, while

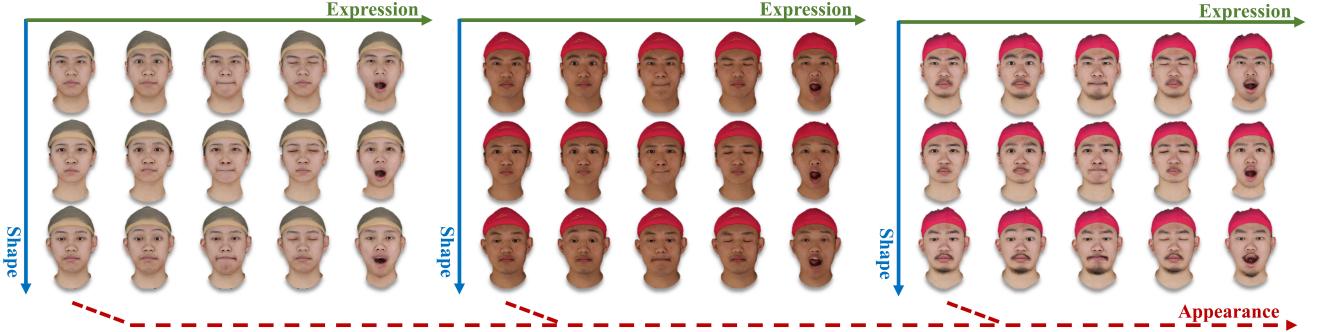


Figure 8. The morphable face is shown by changing parameters in the dimension of shape (blue), expression (green), and appearance (red). We can see that the characteristics in three dimensions are well disentangled.

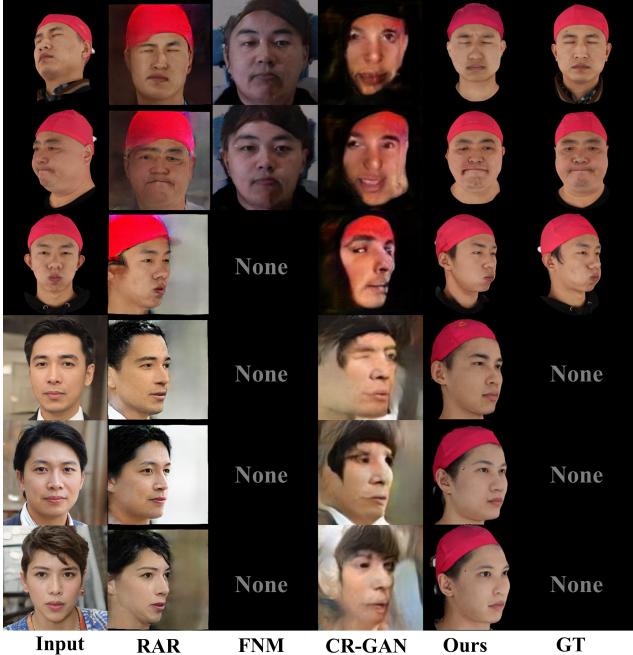


Figure 9. We compare our method with previous face rotation methods. Please note that some results of FNM are ‘None’ because FNM can only rotate from side views to the frontal view.

i3DMM is the learning-based 3DMM that represents shape via SDF. Both models are trained on the same data as described in Section 3.4. The default generated number of parameters for FaceScape is very large( $\approx 630M$ ), so to be fair, we also generated a model with a similar number of parameters to our model( $\approx 120M$ ), labeled as Facescape-equal. PSNR [17], SSIM [52] and LPIPS [56] are used to measure the objective, structural, and perceptual similarity respectively. The better performance in similarity between the generated face image and ground truth indicates better representation ability.

From the visual comparison in Figure 6, we can see that the FaceScape bilinear model doesn’t model pupils and inner mouth, as it is hard to capture accurate 3D geometry for these regions. The rendered texture is blurry due to the

misalignment in the registration phase and the limited representation ability of the linear PCA model. i3DMM is able to synthesize the complete head, but the rendering result is also blurry. We observed that the performance of i3DMM trained on our training set has degraded to some extent, and we think it is because our data amount is much larger than theirs (10 times larger), which makes the task more challenging. By contrast, our model yields the clearest rendering result, which is also verified in quantitative comparison shown in Table 1. The refinement improves the LPIPS but decrease PSNR and SSIM, we believe this is because the GAN loss and perceptual loss focus on hallucinate plausible details but is less faithful to the original image.

## 5.2. Disentanglement Evaluation

We show the synthesis results of different parameters in Figure 8 to demonstrate that identity shape, expression shape, appearance are well disentangled, and shown the interpolation of different attributes in Figure 5, to demonstrate that the face can morph continuously.

## 5.3. Ablation Study

We design several experiments by replacing a certain module with other options to evaluate the effectiveness:

- (a) Replace the learnable expression code with one-hot code for 20 expressions;
- (b) Replace the learnable expression code with the expression parameter borrowed from the FaceScape model;
- (c) Replace the shape code borrowed from the FaceScape model with one-hot code for 300 subjects;
- (d) Replace the shape code borrowed from the FaceScape model with the learnable code;
- (e) Remove the input appearance code, so both the shape and appearance information are memorized in shape code;
- (f) Remove ISM to make the  $\epsilon$  directly fed to network;
- (g) Replace landmark-based sampling strategy with uniform sampling strategy in the training phase;
- (h) Ours with no changes.

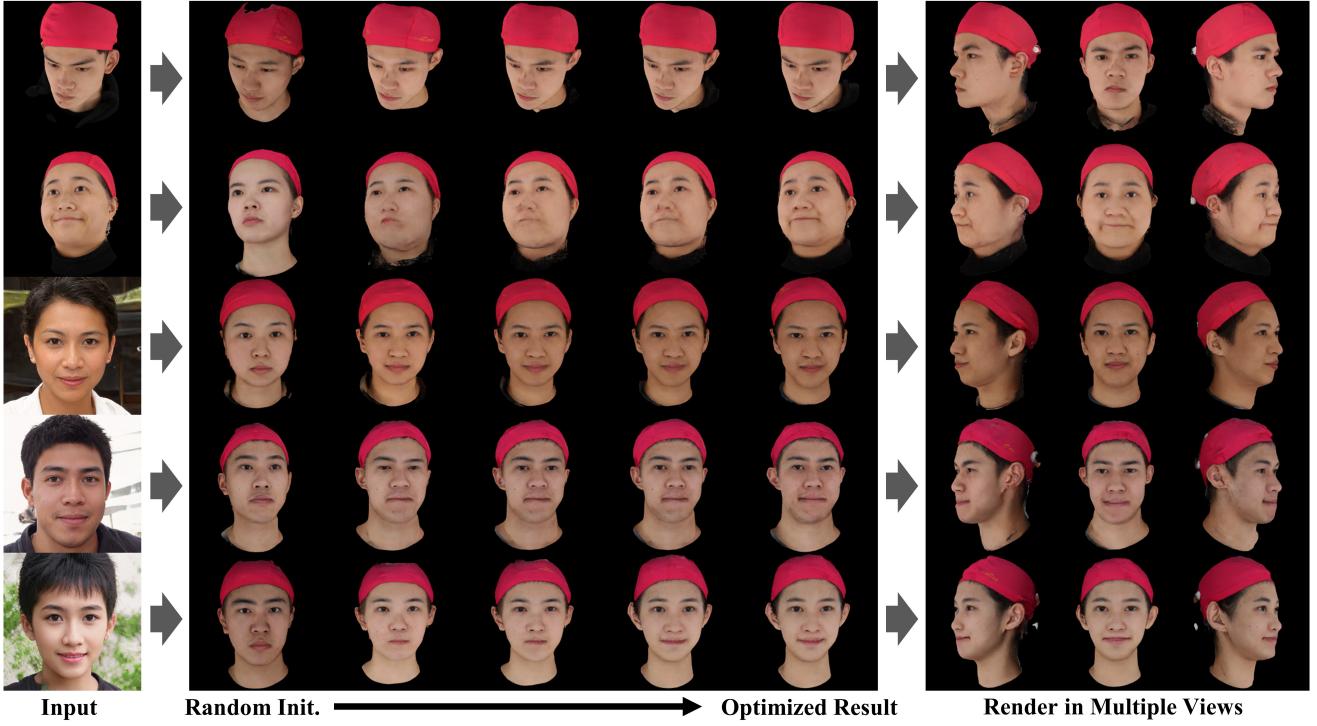


Figure 10. The fitting results to a single-view image of MoFaNeRF. The testing image are from FaceScape testing set and in-the-wild images. The comparison with single-view face reconstruction and failure cases is shown in the appendix.

We reconstruct 300 images of the first 15 subjects in our training set for evaluation, with random view directions and expressions. The results are reported quantitatively in Table 2 and qualitatively in Figure 7. Comparing (h) to (a) and (b), we find the PCA identity code most suitable for encoding shape, which reflects the shape similarity in the parameters space. Comparing (h) to (c) and (d), we can see that learnable code is most suitable for encoding expressions. We think the reason is that the categories of the expression are only 20, which is quite easy for the network to memorize and parse, while PCA code doesn't help for the few categories. Comparing (e), (f), (g) to (h), we can see that our proposed TEM, ISM, and landmark-based sampling all have positive effects on model representation ability, and synthesize more faithful results in the visual comparison.

#### 5.4. Application Result

We show the results of several applications. The description about these applications are explained in Section 4.

**Image-based Fitting.** We show the fitted result to the testing set and in-the-wild images in Figure 10. More results, comparison with single-view reconstruction methods, and failure cases can be found in the appendix.

**View extrapolation.** We fit MoFaNeRF to a single image, then synthesize the images in the novel view to achieve view extrapolation. The results of our method and RAR [58], FNM [38], CR-GAN [43] are shown in Figure 9. We can

see that our method generates the most plausible results for facial regions.

**Face rigging and editing.** As shown in Figure 1, after the model is fitted or generated, we can rig the face by driving the expression code  $\epsilon$ , and edit the face by changing shape code  $\beta$  and appearance code  $\alpha$ . Please watch our results in the video and appendix.

## 6. Conclusion

In this paper, we propose MoFaNeRF that is the first facial parametric model based on neural radiance field. The model disentangles the shape, appearance, and expression to make the face morphable in a large-scale solution space. MoFaNeRF can be used in multiple applications achieving competitive performance comparing to SOTA methods.

**Limitation.** Our model doesn't explicitly generate 3D shapes, and only focuses on rendering performance. Though 3D shapes can be extracted from the neural radiance field by some means, the 3D accuracy is unwarranted. Besides, the single-view fitting of MoFaNeRF only works well for relatively diffused lighting, while the performance will degrade for extreme lighting conditions.

**Ethic discussion.** The use of human face images in this paper has been authorized. Our method may be abused to raise ethical issues that need to be addressed. Please refer to the appendix for the proof and discussion.

**Acknowledgement.** We thank Dr. Yao Yao for his valuable

suggestions on this work and Dr. Yuanxun Lu for proof-reading our paper.

## 6.1. Appendix

### 6.2. Animation of Rigging and Editing

After the face is fitted or generated, it can be rigged driving the expression code  $\epsilon$ , and be edited by changing shape code  $\beta$  and appearance code  $\alpha$ . The animation of rigging and editing results are shown in the supplementary video (Part 3 and 4). We can see that the face can morph smoothly in the dimensions of shape, appearance and expression.

### 6.3. Fitting v.s. Single-View Reconstruction

As shown in Figure 11, we compare our method with four state-of-the-art Single-View Reconstruction(SVR) methods [10, 16, 41, 53] in rendering performance. These methods take the single-view image as input and predict the mesh with texture. The images are rendered from the predicted meshes in the frontal view and  $\pm 60$  side views. Please note that FaceScape-fit [53], 3DDFAv2 [16] and DECA [10] reconstructed the full head, however, their textures come from the source image and only facial textures are assigned. Therefore, we only render the regions with texture for these three methods.

We can see that the inaccurately predicted shape of FaceScape-fit [53], 3DDFAv2 [16] and DECA [10] leads to the artifacts in the side views, as shown in the red dotted circles. Besides, these methods commonly contain wrong scratches on the cheeks due to the misalignment of the predicted shape and the source image. Though the MGCNet doesn't contain the scratches problems, its texture tends to be a mean texture with less detail. We can also observe that in some cases the shape of the nose is unfaithful. Besides, all four methods cannot align ears well, so no texture is assigned to ears. By contrast, our rendering results contain fewer artifacts and are more plausible in the side views. The ears are also rendered in our method, which makes the side-view rendering complete.

### 6.4. More Results of image-based fitting

We show more faces fitted to a single image in Figure 12, which is the extension of Figure 10 in the main paper.

### 6.5. Failure Cases in Fitting

We show some failure cases of the image-based fitting in Figure 13. The first row shows that the fitting results are bad in the extreme lighting. As our model is trained in the images with relatively diffused lighting, large areas of shadow will interfere with our fitting. Lighting models may be introduced in future work to improve the generalization ability for complex lighting conditions. The second row shows that the fitting may fail for the faces with large

occluded regions. Fitting MoFaNeRF to an occluded face is still a challenging task to be solved. The third row shows that the fitting results degraded for the faces that are quite different from the FaceScape dataset in shape (left) or skin color (right). The generalization of image-based fitting still needs to be improved.

### 6.6. More Results of Random Generation

We show more randomly generated faces in Figure 16, which is the extension of Figure 3 in the main paper.

### 6.7. Parameters of Network

The parameters of our network are shown in Figure 14. The boxes represent the full connection layer, where the numbers represent the number of neurons. The circles with C inside represent the concatenating operation between tensors. The numbers follow the parameter means the dimension of this parameter.  $\alpha, \beta, \epsilon$  are the parameters of appearance, shape and expression.  $\gamma(\mathbf{x})$  and  $\gamma(\mathbf{d})$  mean the position encoding of the position code  $\mathbf{x}$  and viewing direction  $\mathbf{d}$ .  $\sigma$  and  $\mathbf{c}$  are the density and color that form the radiance field. The parameters of TEM module is shown in Table 3.

### 6.8. Details of Training and Data

**Implement details.** Our model is implemented on PyTorch [1]. In our experiment, 1024 rays are sampled in an iteration, each with 64 sampled points in the coarse volume and additional 64 in the fine volume. The strategy of hierarchical volume sampling is used to refine coarse volume and fine volume simultaneously, as defined in NeRF [31]. The resolution of the images rendered by MoFaNeRF is 256, and the RefineNet takes the image rescaled to  $512 \times 512$  as input, and synthesizes the final image in  $512 \times 512$ .

We use the Adam optimizer [23] with the initial learning rate as  $5 \times 10^{-4}$ , decayed exponentially as  $2 \times 10^{-5}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-7}$ . Our model is trained for roughly  $400k$  iterations, which takes about 2 days on dual NVIDIA GTX3090 GPUs.

**Details about data preparation.** Taking face orientation as the reference direction, we evenly select 6 pitch angles from  $-30^\circ$  to  $45^\circ$  and 20 yaw angles from  $-90^\circ$  to  $+90^\circ$  degrees for rendering in 120 viewpoints. The samples of 120 views are shown in Figure 15. We use all the training data to train the network of MoFaNeRF, and randomly select 24,000 rendering results to train the RefineNet.

Initially, we plan to use the raw scanned multi-view images released by FaceScape [53], however, we find the camera locations are not uniform for all these 7180 tuples of images. We contacted the authors of FaceScape and learned that the reason was that the capturing took place in two locations, where the camera setups and parameters were changed several times. Therefore, we use the multi-view images to color the raw scanned models, then render them



Figure 11. We compare our fit and rendering result with SOTA single-view reconstruction methods (SVR). In the red circles, we can see that the inaccurately predicted shape of the nose leads to artifacts in the side view. 3DDFAv2, FaceScape-fit, and DECA commonly contain artifacts on the cheeks due to the misalignment of the predicted shape and the source image. Besides, all four methods cannot align ears well, so no texture is assigned to ears.

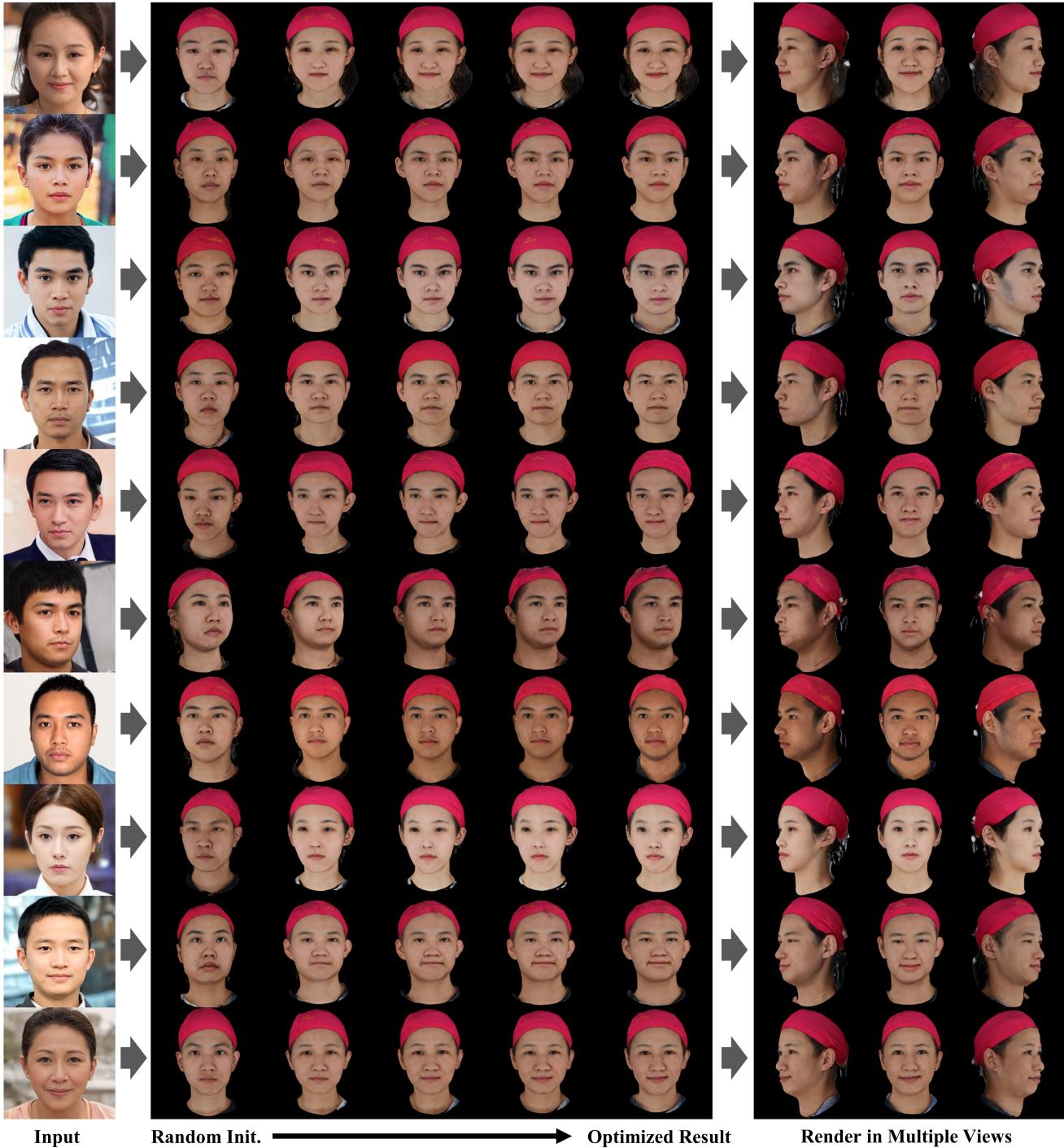


Figure 12. More fitting results of MoFaNeRF to a single-view image.

according to the viewpoints as set above to obtain high-fidelity multi-view images with uniform camera parameters.

### 6.9. Ethic Discussion.

Our method can take a portrait of any individual as input, generate a free-view portrait that can achieve appear-

ance editing and expression rigging. These applications may lead to the infringement of portrait rights. Therefore, we suggest that the protection of individual portrait rights should be emphasized in the applications of this method. It is forbidden to release individual multi-view portraits without authorization and to maliciously vilify and smear por-

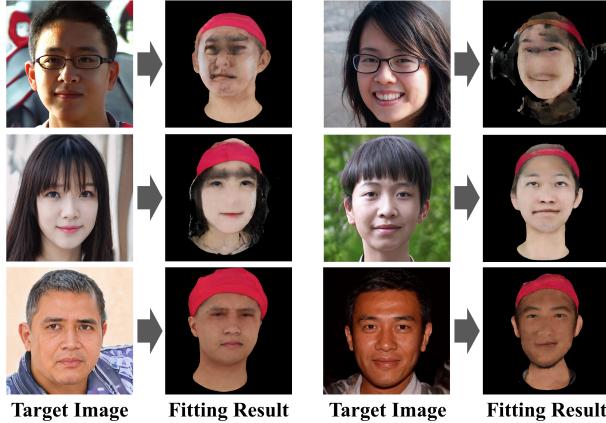


Figure 13. Failure cases of fitting MoFaNeRF to a single image.

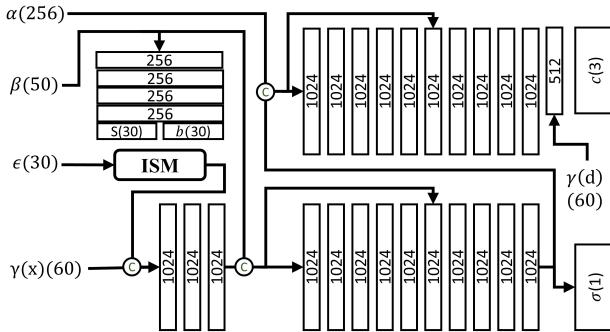


Figure 14. The detailed parameters of the network in MoFaNeRF. The number in brackets indicate the length of the tensor.

traits. In addition, this parameterized model contains the portraits of the collected individuals, which need to be encrypted to protect the portrait information. Due to the limitation of dataset, the experiments in this paper cannot cover various races. This problem can be solved by collecting diverse data of facial images in the future work.

## 6.10. Data Source and License.

The in-the-wild images in Figure 1, 4, 9, 10 are synthesized by StyleGANv2 [21], which is released under Nvidia source code license. So the use of these virtual portraits will not raise infringement issues. We have signed the license agreement with the authors of FaceScape [53] to obtain the permission to use the dataset for non-commercial research purpose, and the permission to publish the subjects of 12, 17, 40, 49, 57, 92, 97, 168, 211, 212, 215, 234, 260, 271, 326 in Figure 2, 5, 6, 7, 8, 9, 10 of this paper.

## References

- [1] Pytorch. <https://pytorch.org/>. 9
- [2] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling facial geometry using compositional vaes. In *CVPR*, pages 3877–3886, 2018. 2

Table 3. The detailed parameters of the TEM in MoFaNeRF. All convolution layers and linear layers are followed by Leaky ReLU [30] with negative slope of 0.2 and 0.1 respectively, except for layers ‘mu’ and ‘std’. ‘Repara.’: means the reparameterization method [24, 51] to produce latent code from the distribution  $\mathcal{N}(\mu, \sigma)$ .  $k$ : kernel size ( $k \times k$ ).  $s$ : stride in both horizontal and vertical directions.  $p$ : padding size ( $p \times p$ ).  $c$ : number of output channels.  $d$ : output spatial dimension ( $d \times d$ ). ‘Conv’: convolution layer. ‘Linear’: fully connected layer. ‘Flatten’: flatten layer.

Name	Type	input	(k,s,p)	c	d
conv1	Conv	textureMap	(4,2,1)	32	256
conv2	Conv	conv1	(4,2,1)	32	128
conv3	Conv	conv2	(4,2,1)	32	64
conv4	Conv	conv3	(4,2,1)	32	32
conv5	Conv	conv4	(4,2,1)	64	16
conv6	Conv	conv5	(4,2,1)	128	8
conv7	Conv	conv6	(4,2,1)	256	4
flat0	Flatten	conv7	–*	4096	1
line1	Linear	flat0	–	512	1
mu	Linear	line1	–	256	1
logstd	Linear	line1	–	256	1
para.	Repara.	(mu,logstd)	–	256	1
line2	Linear	para.	–	256	1
line3	Linear	line2	–	256	1
app.	Linear	line3	–	256	1

\* ‘–’ means meaningless parameters.

- [3] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, volume 99, pages 187–194, 1999. 1, 2
- [4] Chen Cao, Yanlin Weng, Shun Zhou, Yiyang Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 20(3):413–425, 2013. 2
- [5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021. 1, 2
- [6] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular rgb videos. *arXiv preprint arXiv:2106.13629*, 2021. 2
- [7] Shiyang Cheng, Michael Bronstein, Yuxiang Zhou, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. Meshgan: Non-linear 3d morphable models of faces. *arXiv preprint arXiv:1903.10384*, 2019. 2
- [8] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *NIPS*, 29:658–666, 2016. 5
- [9] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ToG*, 39(5):1–38, 2020. 2



Figure 15. Multi-view images are generated in 120 views with 6 pitch angles in  $[-30^\circ \sim +45^\circ]$  and 20 yaw angles in  $[-90^\circ \sim +90^\circ]$ .



Figure 16. More random generated results by our models. We visualize them in three views with yaw angles in  $[-60^\circ, 0^\circ, 60^\circ]$  and pitch angles in  $0^\circ$ .

- [10] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ToG(SIGGRAPH)*, 40(4):1–13, 2021. [9](#)
- [11] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *CVPR*, pages 8649–8658, 2021. [1, 2](#)
- [12] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. <https://arxiv.org/abs/2012.05903>, 2020. [2](#)
- [13] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. [5](#)
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 27, 2014. [5](#)
- [15] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenet: A style-based 3d-aware generator for high-resolution image synthesis, 2021. [1, 2](#)
- [16] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense

- face alignment. In *ECCV*, pages 152–168, 2020. 9
- [17] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *ICPR*, pages 2366–2369, 2010. 7
- [18] Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. In *CVPR*, pages 11957–11966, 2019. 2
- [19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 5
- [20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 4
- [21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 4, 12
- [22] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014. 5
- [23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 9
- [24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 12
- [25] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. *arXiv preprint arXiv:1907.11922*, 2019. 5
- [26] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ToG*, 29(4):32, 2010. 6
- [27] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ToG*, 36(6):194, 2017. 2
- [28] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM SIGGRAPH Asia*, 2021. 2
- [29] Ling Luo, Dingyu Xue, and Xinglong Feng. Ehanet: An effective hierarchical aggregation network for face parsing. *Applied Sciences*, 10(9):3135, 2020. 5
- [30] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013. 12
- [31] 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*, pages 405–421, 2020. 1, 2, 3, 9
- [32] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. *arXiv preprint arXiv:2104.03110*, 2021. 2
- [33] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020. 1, 2
- [34] 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. *arXiv preprint arXiv:2106.13228*, 2021. 1, 2, 4
- [35] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable neural radiance fields for human body modeling. *arXiv preprint arXiv:2105.02872*, 2021. 2
- [36] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 2
- [37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021. 2
- [38] Yichen Qian, Weihong Deng, and Jian Hu. Unsupervised face normalization with extreme pose and expression in the wild. In *CVPR*, pages 9851–9858, 2019. 8
- [39] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In *CVPR*, pages 11733–11742, 2021. 2
- [40] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *CVPR*, 2021. 1, 2
- [41] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. In *ECCV*, pages 53–70, 2020. 9
- [42] 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. In *CVPR*, pages 2549–2559, 2018. 2
- [43] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. Cr-gan: learning complete representations for multi-view generation. *arXiv preprint arXiv:1806.11191*, 2018. 8
- [44] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. In *CVPR*, pages 1126–1135, 2019. 2
- [45] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *CVPR*, pages 7346–7355, 2018. 2
- [46] Luan Tran and Xiaoming Liu. On learning 3d face morphable model from in-the-wild images. *PAMI*, 2019. 2
- [47] Edgar Treitschke, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. <https://arxiv.org/abs/2012.12247>, 2020. 4
- [48] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Joivan Popović. Face transfer with multilinear models. *ToG*, 24(3):426–433, 2005. 2
- [49] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, pages 4690–4699, 2021. 2

- [50] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, pages 8798–8807, 2018. [4](#), [5](#)
- [51] Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhofer. Learning compositional radiance fields of dynamic human heads. In *CVPR*, pages 5704–5713, 2021. [1](#), [2](#), [12](#)
- [52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. [7](#)
- [53] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *CVPR*, 2020. [2](#), [3](#), [6](#), [9](#), [12](#)
- [54] 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 *CVPR*, pages 12803–12813, 2021. [2](#), [6](#)
- [55] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. [2](#)
- [56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. [7](#)
- [57] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *CVPR*, pages 7194–7202, 2019. [5](#)
- [58] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *CVPR*, pages 5911–5920, 2020. [8](#)
- [59] Hao Zhu, Haotian Yang, Longwei Guo, Yidi Zhang, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: 3d facial dataset and benchmark for single-view 3d face reconstruction. *arXiv preprint arXiv:2111.01082*, 2021. [2](#)