LC-NeRF: Local Controllable Face Generation in Neural Radiance Field

Wen-Yang Zhou, Lu Yuan, Shu-Yu Chen, Lin Gao, *Member, IEEE*, and Shi-Min Hu, *Senior Member, IEEE*

***Abstract*—3D face generation has achieved high visual quality and 3D consistency thanks to the development of neural radiance fields (NeRF). However, these methods model the whole face as a neural radiance field, which limits the controllability of the local regions. In other words, previous methods struggle to independently control local regions, such as the mouth, nose, and hair. To improve local controllability in NeRF-based face generation, we propose LC-NeRF, which is composed of a *Local Region Generators Module (LRGM)* and a *Spatial-Aware Fusion Module (SAFM)*, allowing for geometry and texture control of local facial regions. The *LRGM* models different facial regions as independent neural radiance fields and the *SAFM* is responsible for merging multiple independent neural radiance fields into a complete representation. Finally, LC-NeRF enables the modification of the latent code associated with each individual generator, thereby allowing precise control over the corresponding local region. Qualitative and quantitative evaluations show that our method provides better local controllability than state-of-the-art 3D-aware face generation methods. A perception study reveals that our method outperforms existing state-of-the-art methods in terms of image quality, face consistency, and editing effects. Furthermore, our method exhibits favorable performance in downstream tasks, including real image editing and text-driven facial image editing.**

***Index Terms*—3D face generation, neural radiance fields, semantic manipulation.**

I. Introduction

REALISTIC face image generation and manipulation are important topics in image synthesis, finding extensive application in portrait generation and artistic creation. Considerable efforts [1], [2], [3] have been dedicated to enhancing the quality and resolution of generated face images. Simultaneously, there is a growing demand from users to have increased interaction and control over the generated images. In order to enhance the controllability of the generation process, various methods have been proposed that allow for image editing through different interfaces, including sketches [4], texts [5], semantic masks [6], [7], [8], and more.

Benefiting from the implicit 3D representation of neural radiance fields (NeRF) [9], the image synthesis models have shown significant progress in transferring 2D image generation task [1] to 3D [10], [11], [12], addressing 3D consistency in perspective transformation. EG3D [10], StyleNeRF [12], and StyleSDF [11] use implicit three-dimensional representations to improve the quality of 3D face generation. These methods have achieved good geometric quality, but their overall neural radiance field limits the controllability of the local regions.

Recently, some NeRF-based face editing methods [13], [14], [15] have shown excellent results in decoupling the geometry and texture of faces. FENeRF [13], IDE-3D [14] and NeRF-FaceEditing [15] decouple geometry and texture by using separate geometry and texture networks. These methods model the face as a unified Neural Radiance Field (NeRF) and utilize a single global latent code to generate the face. This modelling strategy does not support the direct manipulation of the latent code to selectively control local facial regions, revealing an absence of local decoupling. In the pursuit of editing facial geometry, these methods necessitate the optimization of an optimal global latent code. The purpose is to produce a mask that matches the edited mask, while also ensuring the non-edited areas are maintained as invariant as possible. Nonetheless, optimizing a latent code that can simultaneously meet the demands of successful editing and stability of non-edited areas presents a considerable challenge. Consequently, our method's primary contribution is the decoupled control of the 3D face, enhancing the local controllability of the 3D face and the stability of non-edited regions. Under the condition that each object is independent, spatially diverse, and can appear at varying locations, GIRAFFE [16] can decouple a set of objects. Due to the lack of semantic guidance, GIRAFFE [16] can not separate the local regions of the face.

To improve the local controllability of NeRF-based face generation methods, we propose a semantically guided local controllable face generation method, LC-NeRF, for fine-grained facial local region control and the decoupling of geometry and texture. There are two core issues, one is the decomposition of

Manuscript received 7 March 2023; revised 24 June 2023; accepted 1 July 2023. Date of publication 17 July 2023; date of current version 1 July 2024. This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0112902, in part by the Natural Science Foundation of China under Grant 62220106003, in part by the Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology. Recommended for acceptance by M. H. Kim. *(Corresponding author: Shi-Min Hu.)*

Wen-Yang Zhou and Shi-Min Hu are with BNRist, Tsinghua University, Beijing 100084, China (e-mail: zhouwy19@mails.tsinghua.edu.cn; shimin@tsinghua.edu.cn).

Lu Yuan is with the Department of Computer Science, Stanford University, Stanford, CA 94305 USA (e-mail: luyuan@stanford.edu).

Shu-Yu Chen and Lin Gao are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100045, China (e-mail: chenshuyu@ict.ac.cn; gaolin@ict.ac.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2023.3293653, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2023.3293653

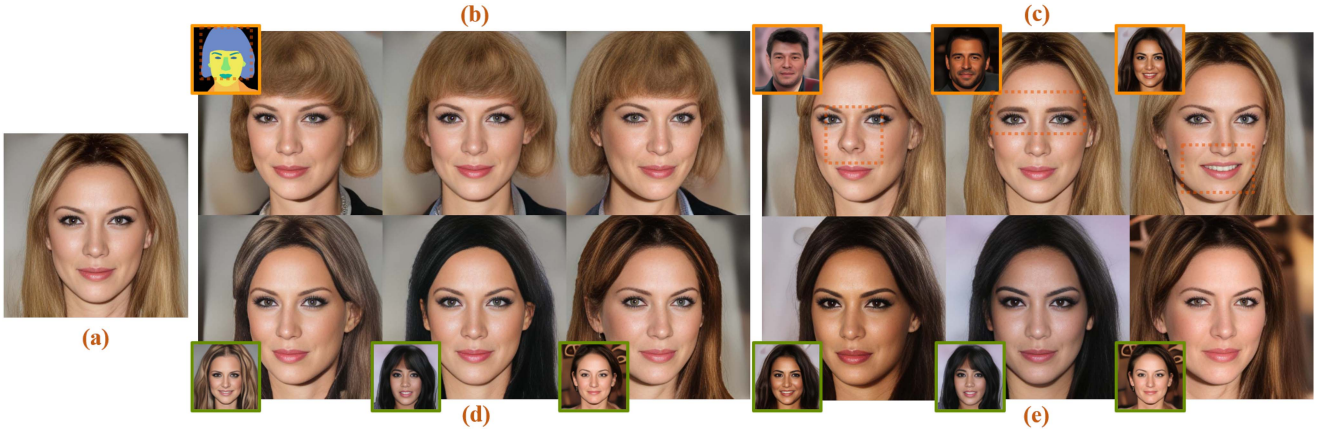

Fig. 1. Given an input image (a) and a reference face image, our method can independently manipulate the geometry of any local region, such as hair, nose, mouth, eyebrows, etc. We show these local and global face editing tasks achieved with our method: (b) manipulating the geometry of the hair by modifying the semantic mask while retaining the geometry of other regions and 3D consistency; (c) manipulating the geometry of more local regions, such as nose, eyebrows and mouth; (d) manipulating the local texture of the hair while retaining the geometry and 3D consistency; (e) manipulating the global texture while retaining the geometry and 3D consistency.

the global 3D representation and representations of the local 3D regions, and the other is the fusion of local 3D regions. The first problem is how to decompose a complete 3D representation into multiple local 3D representations and stably complete the training process. To overcome this issue, we design our generator network with multiple local generators to generate the content for each local region. In addition, for more flexible control over geometry and texture, we further subdivide the local generator into a geometry network and a texture network controlled by geometry code and texture code separately. Through these designs, our method can modify the geometry and texture of local regions without affecting other regions by editing multiple local latent codes. The other core challenge is how to fuse local 3D representations of all local regions to generate the final face image. We propose a *Spatial-Aware Fusion Module* to complete the fusion of multiple local regions. Specifically, for every spatial point sampled, each local geometry generator predicts its semantic confidence and geometry feature, while each local texture generator predicts its texture feature. Subsequently, the fusion module fuses the predicted geometry/texture features of this point. Finally, we leverage the fused features to predict density, semantic mask, and color using fully connected layers. Subsequently, we employ volume rendering operation to obtain the ultimate facial image and semantic mask.

Qualitative and quantitative experiments show that our method enables finer-grained local control than state-of-the-art methods. The main contributions of this paper are summarized as followed:

- We introduce LC-NeRF, a semantically guided local controllable NeRF face generation method that enables decoupled control over the geometry and texture of local facial regions.
- We propose a *Local Region Generators Module* to decompose the global 3D representation and latent codes into multiple local regions, and a *Spatial-Aware Fusion Module* that aggregates these regions into a whole image.
- Our method achieves finer-grained local control than state-of-the-art 3D face generation methods. Both qualitative and quantitative evaluations prove the effectiveness of our method in local control.

II. Related Work

A. Neural Face Generation

Generative models [17], such as Stylegan v1-v3 [1], [2], [3], have achieved high-qulity generaton of 2D images. In recent years, NeRF [18] has emerged as a method that can implicitly model 3D geometry from 2D images and then render photorealistic and 3D consistent images. Subsequently, NeRF-based face generative models have been investigated. PI-GAN [19] proposes a SIREN-based [20] implicit radiance field to generate 3D faces via sampled latent and positonal encoding. Furthermore, due to the advantages of StyleGAN [1] in image generation, some methods [11], [12] based on StyleGAN can generate high resolution and quality images. StyleNeRF [12] provides a 3D GAN approach that fuses style-based generation with scene representation by neural radiance fields. StyleSDF [11] is similar, but incorporates an SDF-based 3D representation to ensure that images generated from different viewpoints have 3D geometric consistency. In addition, some methods study different forms of space representation. For example, EG3D [10] uses three projection planes (tri-plane) to represent the 3D space, generated by a backbone of StyleGAN. GRAM [21] proposes a radiance manifolds based generative model that divides the space into multi-manifolds. These methods improves the quality of generated images but lack the editability and controllability of geometry and texture. Our method enhances the effect of disentanglement of facial features while maintaining generative quality.

B. Neural Face Decoupling

The rapid progress in image generation models has enabled the generation of high-quality portraits. Meanwhile, more and more methods focus their attention on face decoupling. The scope of face decoupling tasks has expanded beyond the 2D

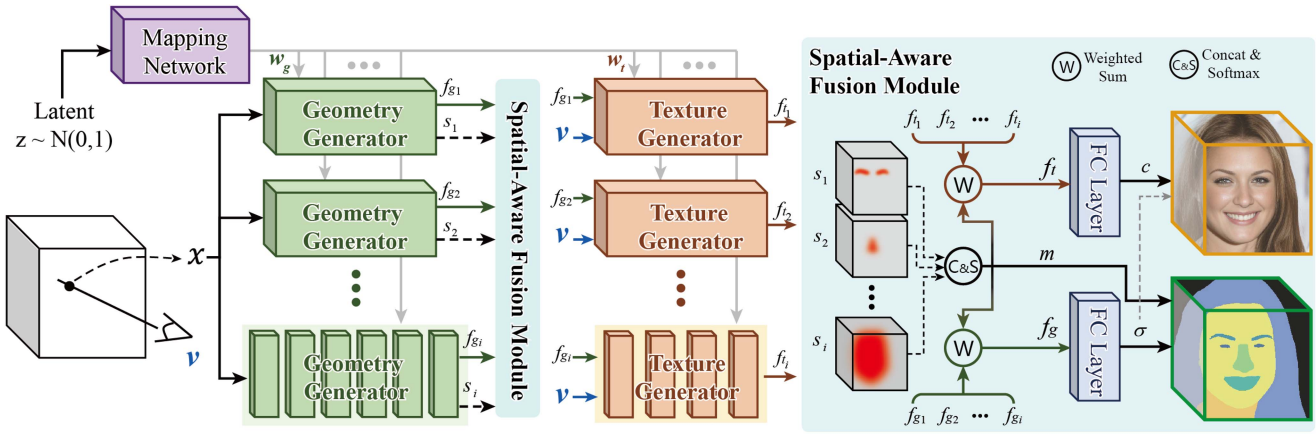

Fig. 2. Pipeline of our framework LC-NeRF. Our pipeline is composed of multiple local generators and a spatial aware fusion module. The local generators include geometry and texture generators, separately controlled by geometry latent code w_g and texture latent code w_t. LC-NeRF can modify the geometry or texture of an local region directly by editing its latent code w_g or w_t.

domain [6], [22], and there is a growing interest in exploring decoupling and control techniques for 3D faces [13], [14], [15], [23]. DeepFaceEditing [4] decouples facial local regions by using sketches to represent geometry. SofGAN [23] trains a semantic occupancy field (SOF) and uses 2D semantic masks to generate face images to decouple geometry and texture. SemansticStyleGAN [6] enhances the control over local regions by generating the features of each region separately and then fusing the features of different regions in the 2D feature domain.

Inspired by the implicit 3D representation and generation techniques in NeRF, research on 3D face decoupling has gained significant momentum. FENeRF [13] introduces a mask branch to PI-GAN [19] to enhance geometry control. Furthermore, IDE-3D [14] and NeRFFaceEditing [15] achieve the decoupled control of geometry and texture by leveraging three projection planes [10]. IDE-3D [14] proposes a framework with separate geometry and texture networks to generate respective tri-plane features. Inspired by AdaIN, NeRFFaceEditing [15] decomposes the tri-plane features into geometry features and appearance features for decouping the geometry and appearance. Despite the successes of these 3D-aware face generation methods [13], [14], [15] in decoupling the geometry and texture of faces and yielding commendable results, they still lack the ability to exert fine-grained control over local regions. In contrast, our proposed method leverages the supervision of semantic masks to achieve more precise decoupling and manipulation of the human face.

III. Methodology

In this section, we present a comprehensive overview of the architecture and details of our method. The objective of LC-NeRF is to enable independent control over the geometry and texture of local regions, encompassing areas such as the mouth, nose, hair, and background. To accomplish this, we introduce the *Local Region Generators Module*, which consists of separate lightweight local networks for each local region: a geometry network and a texture network. These networks are controlled by geometry and texture latent codes, respectively (Section III-A). To integrate the local facial regions into a whole face, we devise a *Spatial-Aware Fusion Module*. This module facilitates the fusion of features generated by all the local generators and subsequently generates the final facial image using volume rendering operation (Section III-B). Furthermore, we introduce the two discriminators in our framework, accompanied by a comprehensive description of the employed loss functions for network training (Section III-C).

A. Local Region Generators Module (LRGM)

1) Local Geometry Generator: To individually control the geometry of local regions, we assign a lightweight geometry generator Φs_i for each local region i of the face. If a 3D point belongs to a certain local region, the corresponding geometry generator provides the most information for this point. The generator plays a major role in determining the semantic category and geometry information of the point. As shown in Fig. 2, each geometry generator contains 6 linear layers with SIREN [20] activation, controlled by the geometry latent code w_g.

Given a sampled point $x \in \mathbb{R}^3$, the i_{th} geometry generator module Φs_i decodes it to obtain the semantic confidence $s_i(x)$ and geometry feature $f_{g_i}(x)$ from a geometry latent w_{g_i}:

$$s_i(x), f_{g_i}(x) = \Phi s_i(x, w_{g_i}) \tag{1}$$

Here, $s_i(x)$ indicates the probability that the i_{th} local geometry generator believes three-dimensional point x to be in its region. $s_i(x)$ has two characteristics: i) The larger the value of $s_i(x)$, the more importance and more proportion the geometry feature $f_{g_i}(x)$ is; ii) Sampling or modifying the geometry latent w_{g_i} can increase or reduce the $s_i(x)$ value of the local region i, which enables local editing of geometry. Specifically, we use a linear layer following the geometry feature $f_{g_i}(x)$ to calculate the geometry confidence $s_i(x)$.

2) Local Texture Generator: The texture generator can be interpreted as a shader, which is used to fill the color of the geometry generated by the geometry generator. In other words, the

texture generators do not participate in or affect the generation of geometry, and the geometry generator is only used to determine the shape of the face, so as to achieve local region decoupling and geometry/texture decoupling. Each texture generator contains 4 linear layers with SIREN [20] activation, and is controlled by the texture latent code w_t.

Given the viewing direction $v \in \mathbb{R}^3$, the i_{th} local texture generator module Φ_{t_i} decodes the texture feature $f_{t_i}(x)$ from a geometry latent w_{t_i}:

$$f_{t_i}(x) = \Phi_{t_i}(f_{g_i}(x), v, w_{t_i}) \tag{2}$$

The texture features predicted by all the local texture generators will be fused in subsequent fusion module, and the final color value of the sampled 3D point x will be predicted.

B. Spatial-Aware Fusion Module (SAFM)

The spatial-aware fusion module is designed for interaction and aggregation among multiple local generators. The proposed fusion module fuses the features of different generators with a soft and adjustable mechanism and generates the whole image. We concatenate the semantic confidence $s_i(x)$ of all the geometry generators and apply the softmax activation to obtain the semantic mask $m(x)$.

$$m_i(x) = \frac{e^{s_i(x)}}{\sum_{k=1}^{K} e^{s_k(x)}} \tag{3}$$

where K is the number of local regions. We use the semantic mask $m(x)$ to fuse the geometry features $f_{g_i}(x)$ to get the final geometry feature $f_g(x)$.

$$f_g(x) = \sum_i (m_i(x) * f_{g_i}(x)) \tag{4}$$

$f_g(x)$ is the geometry feature of the 3D point extracted by our proposed geometry generators. We use a linear layer after $f_g(x)$ to predict the signed distance field (SDF) value $d(x)$ of the 3D point x. Then we convert the SDF value to volume density $\sigma(x)$ by the following formula [11].

$$\sigma(x) = Sigmoid(d(x)/\beta)/\beta \tag{5}$$

where β is a learnable parameter. The smaller β is, the more the volume density $\sigma(x)$ will converge on the surface of the face. In our experiments, the initial value of $\sigma(x)$ is set to 0.1. As the training progresses, the value of β will become smaller and smaller.

The texture features $f_{t_i}(x)$ are also fused with the semantic mask $m(x)$ to get the final texture feature $f_t(x)$. And then we use one linear layer after $f_t(x)$ to get the color value $c(x)$:

$$f_t(x) = \sum_i (m_i(x) * f_{t_i}(x)) \tag{6}$$

We render the generated image I' and the generated semantic mask M' through the volume rendering. Given a camera position o, by shooting a ray $r(t) = o + tv$ at each pixel, we calculate the color and mask of N points sampled from t_n to t_f on the ray. In our experiments, N is set to 18.

$$\begin{aligned} I'(r) &= \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), v)dt, \\ M'(r) &= \int_{t_n}^{t_f} T(t)\sigma(r(t))m(r(t), v)dt, \\ \text{where } T(t) &= \exp\left(-\int_{t_n}^{t} \sigma(r(s))ds\right) \end{aligned} \tag{7}$$

At this point, we complete the fusion operation through *Spatial Aware Fusion Module* to generate the whole image I' and the semantic mask M'.

C. Discriminators and Loss Function

In order to ensure quality of the generated image and correspondence between the image and the mask, we propose a double discriminator supervision strategy. ~~One~~The first discriminator is the image quality and pose awared discriminator D_I, which is used to distinguish between real images and generated images and ~~predicts~~to predict the azimuth and the elevation θ'. In addition to the GAN loss [17], we use a smoothed L1 loss $\mathcal{L}_{pose}$ and R1 regularization loss to supervise the training of D_I for the generated images.

$$\begin{aligned} \mathcal{L}_{D_I} &= \mathbb{E}[1 + exp(D_I(I'))] + \mathbb{E}[1 + exp(-D_I(I))] \\ &\quad + \lambda_{I_{reg}}\mathbb{E}\|\nabla D_I(I)\|^2 + \lambda_{pose}\mathcal{L}_{pose}(\theta, \theta') \end{aligned} \tag{8}$$

where I' and M' are the fake image and the semantic mask generated by LC-NeRF with the sampled pose θ. I and M are the ground truth image and the mask sampled from the real dataset. where $\lambda_{I_{reg}}$, λ_{pose} are set to 10 and 15 respectively.

The other discriminator is the image and semantic mask discriminator D_{IM}, which is used to determine whether the image is consistent with the semantic mask. We also regularize the gradient norm for this discriminator with R1 regularization loss.

$$\begin{aligned} \mathcal{L}_{D_{IM}} &= \mathbb{E}[1 + exp(D_{IM}(I', M'))] \\ &\quad + \mathbb{E}[1 + exp(-D_{IM}(I, M)] \\ &\quad + \lambda_{IM_{reg}}\mathbb{E}\|\nabla D_{IM}(I, M)\|^2 \end{aligned} \tag{9}$$

where $\lambda_{IM_{reg}}$ is set to 10.

The generator G is supervised by the two discriminators D_M and D_{IM} and the camera pose loss L_{pose}. In addition, we also introduce geometry supervision of SDF with eikonal loss [18] and minimal surface loss [11].

$$\begin{aligned} \mathcal{L}_G &= \mathbb{E}[1 + exp(-D_I(I'))] \\ &\quad + \lambda_{IM}\mathbb{E}[1 + exp(-D_{IM}(I', M'))] \\ &\quad + \lambda_{\text{pose}}\,\mathcal{L}_{\text{pose}}\,(\theta, \theta') + \lambda_{eik}\mathbb{E}[\|\nabla d(x)\|_2 - 1]^2 \\ &\quad + \lambda_{sur}\mathbb{E}[exp(-100|d(x)|] \end{aligned} \tag{10}$$

where λ_{IM}, λ_{pose}, λ_{eik}, λ_{sur} are set to 0.5, 15, 0.1, 0.05 respectively.

Fig. 3. Multi-view face images and semantic masks with a resolution of 512, generated by LC-NeRF trained on CelebAMask-HQ dataset.

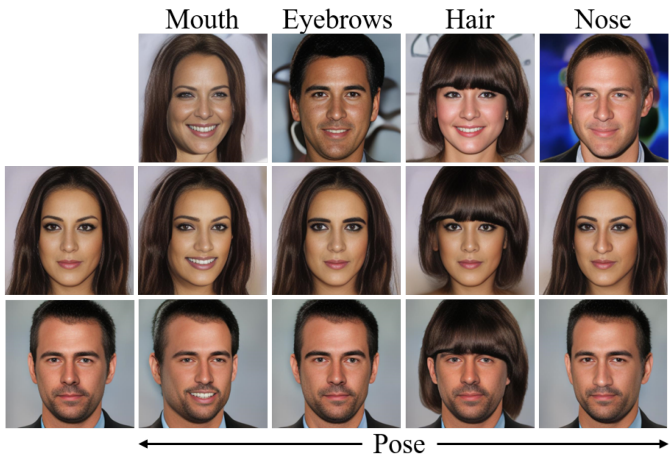

Fig. 4. Results of local style Manipulation. LC-NeRF supports manipulating the geometry and texture of any local region of other faces to the target face. Here we show the results of multi view synthesis that migrate the geometry and texture of an local region at the same time.

TABLE I
QUANTITATIVE METRIC TO SHOW THE EFFECT OF INVERSION

Method	FENeRF	IDE-3D	NeRFFE	Ours
MSE_{inv} (↓)	162.91	35.11	46.09	**19.72**

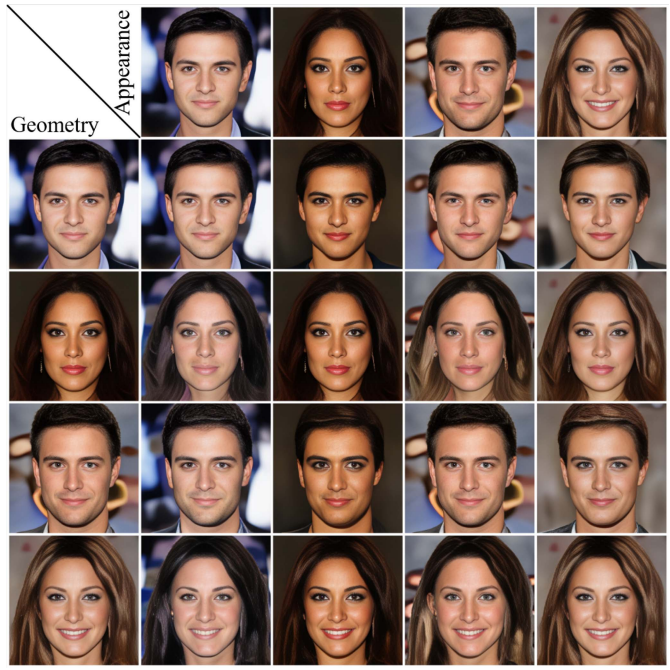

Fig. 5. Results of global style Manipulation. LC-NeRF supports global modification of face texture. The figure shows examples of transferring the texture information of the reference face to the target face.

IV. EXPERIMENTS

In this section, we first introduce our experimental setup (Section IV-A & B). Then we show the results of local and global manipulation, which is a unique function of our method (Section IV-C & D). To demonstrate the effectiveness of LC-NeRF, we compare the manipulation effects on real images with state-of-the-art face editing methods, including FENeRF [13], IDE-3D [14], and NeRFFaceEditing (NeRFFE) [15] (Section IV-E). Furthermore, we evaluate the computational costs of all methods (Section IV-F) and conduct a perception study to assess the effectiveness of LC-NeRF (Section IV-G). Additionally, we conduct an ablation study to verify the effectiveness of the proposed module (Section IV-H) and showcase the application of text-driven face editing (Section IV-I).

A. Training Datasets

We train LC-NeRF on the CelebAMask-HQ dataset [24], which contains 30,000 high-quality face images with 1024×1024 resolution. For this dataset, each image provides an accurate segmentation image with 19 categories. In our

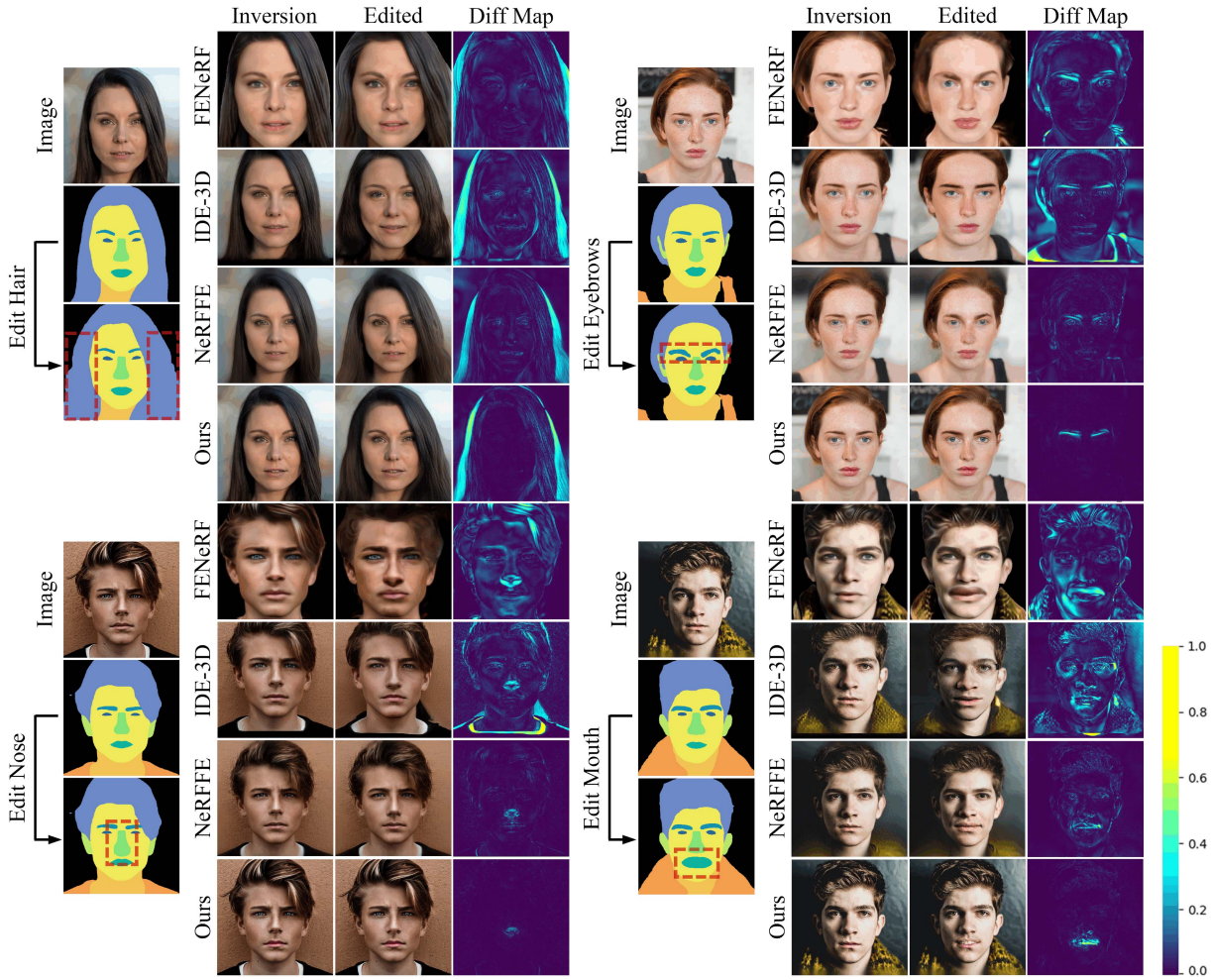

Fig. 6. Comparison of local geometry editing with FENeRF [13], IDE-3D [14], NeRFFE [15]. For each sample, the left side displays the real image, source and edited mask from top to bottom. The right side is inversion results, edited results and difference maps of different methods.

experiments, we combine the left and right local regions into one, such as glasses and eyebrows. After processing, there are 13 types of face local regions.

B. Implementation Details

We use the Adam [25] optimizer with $\beta_1 = 0$ and $\beta_2 = 0.9$ to train the generator and discriminators, and the learning rates of G, D_I, D_{IM} are 0.00002, 0.0002 and 0.0002 respectively. We train LC-NeRF on 8 NVIDIA GeForce 3090 GPUs for 48 hours with a batch size of 24. The resolution of the volume rendering is 64. In order to accomplish local region decoupling, the *Local Region Generators Module* incorporates 13 individual local generators. To maintain a balance between resources and methods, LC-NeRF employs a two-stage training strategy. In the first stage, we train the low-resolution *Local Region Generators Module*. In the second stage, we train a super-resolution network [26] to upscale the low-resolution images to 512x512 resolution. During inference, it only takes 0.10 s to generate a face image and corresponding semantic mask on 1 NVIDIA GeForce 3090 GPU. LC-NeRF is implemented on Jittor [27], a fast-training deep learning framework.

C. Local Style Manipulation

In our framework, we can sample random latent codes to generate both face images and semantic masks. As shown in Fig. 3, our method can generate diverse face images, and the multi-view results prove that our method maintains the 3D consistency across different views. Moreover, we demonstrate the effects of local and global style manipulation achieved by LC-NeRF.

In contrast to existing methods, our approach enables the manipulation of both the geometry and texture of local regions. Modifying the geometry of specific regions is achieved by directly altering the geometry latent code w_g associated with the respective local region, enabling precise local geometry editing. Fig. 4 shows the multi view editing results of modifying mouth, eyebrows, hair, and nose. It can be observed that LC-NeRF can edit target regions accurately while keeping non-editing regions unaffected.

D. Global Style Manipulation

Our method enables global manipulation of both the geometry and texture of the entire face. Specifically, we can modify the global texture of all local regions while keeping the geometry unchanged. This is accomplished by directly modifying the texture latent codes w_t of all the local regions. Similarly, global geometry modification follows the same principle. In Fig. 5, we show examples of transferring styles of reference images to target images. It can be observed that the geometry of all the local regions remains unchanged when the texture is modified, which also verifies the decoupling property of geometry and texture.

E. Real Image Local Manipulation

We can edit the images generated by the latent codes w_g and w_s at a certain pose as well as the real images. To edit the real images, we need to encode the real images into the $\mathcal{W}^+$ [1] space through pivotal tuning inversion [28]. Given a real face image I and the corresponding semantic mask M, we first invert I to generate the latent code w by optimizing $\mathcal{L}_{inv}$.

$$\mathcal{L}_{inv} = \mathcal{L}_{LPIPS}(I, I') + \|M - M'\|_2 + \|\boldsymbol{I} - \boldsymbol{I'}\|_{\boldsymbol{2}} \quad (11)$$

Where I' and M' are the generated face image and semantic mask.

When the user edits the mask and gets the edited mask M_e, our optimization goal is to find an editing vector δw to make the mask M' generated by $\delta w + w$ close to the editing mask M_e. We use the mean square error (MSE) between the edited mask M_e and generated mask M'. During editing, we optimize the geometry latent code of the corresponding local region for 500 iterations.

The most important quality of face editing is to change the target region while ensuring that the non-editing regions are not affected. Otherwise, the edited image may become too dissimilar to the original that it may be interpreted as another person entirely. For fair comparison, all evaluated methods are tested on the II2S [29] dataset without any fine-tuning. The II2S dataset contains 120 high-quality face images with different styles. We use a pretrained face parsing method [30] to extract the same semantic mask for all methods.

1) Local Geometry Editing: Local geometry editing is an interactive and practical application of editing face images by modifying corresponding masks. In comparison, we appropriately increase the learning rate of IDE-3D inversion to ensure that it converges to the best effect. The inversion and local editing results are shown in Fig. 6. By leveraging its inherent local decoupling characteristics, LC-NeRF effectively ensures that the desired modifications are applied specifically to the target regions, while leaving the non-editing regions unaffected. Additionally, the editing results of FENeRF appear unnatural and unrealistic. Moreover, since IDE-3D and NeRFFaceEditing edit images in the global latent space, the edits inevitably affect non-editing regions, resulting in obvious changes outside of the target region. For instance, in the case of editing the mouth, there are obvious modifications made to other regions of the face in the IDE-3D results. FENERF is limited in hair editing because the face occupies most of the image area during inversion. In Fig. 9, we show the results of more complex manipulation on diverse faces.

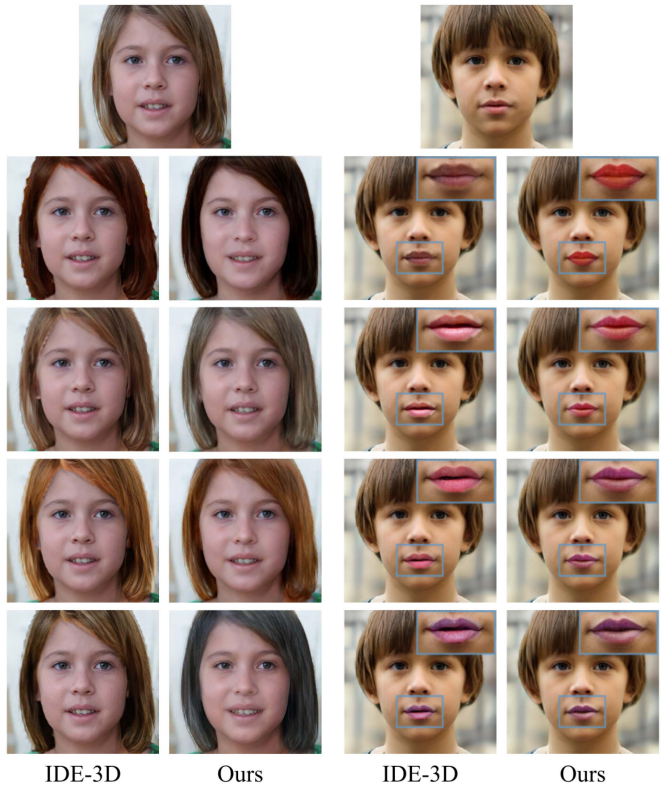

Fig. 7. Qualitative comparison results of real image local texture editing between IDE-3D and LC-NeRF(Ours). Two cases show the editing effects of two methods to modify different hair and lip texture colors.

TABLE II
QUANTITATIVE METRICS TO SHOW THE CORRECTNESS OF EDITING

Method	FENeRF	IDE-3D	NeRFFE	Ours
MSE_{mas} (↓)	182.58	107.28	52.94	**29.96**

We use quantitative metrics to demonstrate the effectiveness of LC-NeRF. The first four metrics are scaled by $1e^{-3}$ for better readability. Let's define the original face image as I_{ori}, the inversion face image as I_{inv}, the edited face image as I_{edi}, the edited mask image as M_{edi}. We use the *Pixel-wise Mean Squared Error (MSE)* as the metric:

To show the effect of inversion, we calculate MSE_{inv} by Formula 12. The results are shown in Table I. LC-NeRF achieves the best inversion effect.

$$MSE_{inv} = MSE(I_{ori}, I_{inv}) \quad (12)$$

To show the correctness of editing, we first predict the mask M_{res} of I_{edi} with a pretrained face parser [30]. Then, we calculate MSE_{mas} by Formula 13. The results are shown in Table II. LC-NeRF completes the manipulation most correctly. All other methods have failed edits. For example, in the fifth and

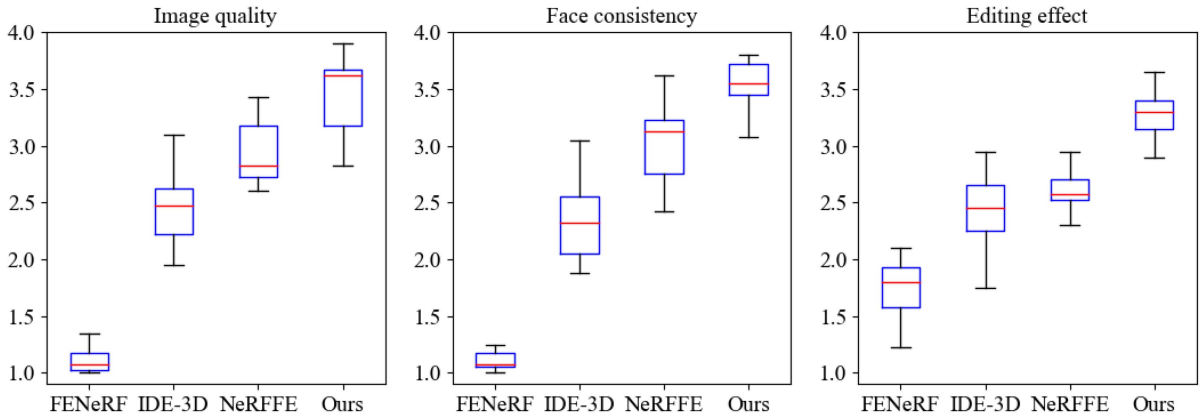

Fig. 8. Box plots of image quality, face consistency, and editing effect, based on the participants in the perception study with four methods: FENeRF [13], IDE-3D [14], NeRFFE [15] and Ours.

TABLE III
QUANTITATIVE METRIC TO DEMONSTRATE THE INVARIANCE OF NON-EDITING REGIONS AND THE EFFECTIVENESS OF DECOUPLING

Method	FENeRF	IDE-3D	NeRFFE	Ours
MSE_{OE} (↓)	154.86	58.76	46.21	**26.15**
MSE_{IE} (↓)	44.08	54.98	23.52	**21.70**

It should be noted that NeRFFE has achieved relatively good results because it has constrained the non-editing regions to remain unchanged during optimization. LC-NeRF can achieve the best results without such constraints.

TABLE IV
WE CAN GENERATE AN IMAGE IN A REASONABLE TIME EVEN THOUGH OUR TASKS ARE MORE COMPLEX AND FINE-GRAINED

Method	FENeRF	IDE-3D	NeRFFE	Ours
Resolution	256x256	512x512	512x512	512x512
Time (↓)	1.21 s	0.03 s	0.05 s	0.10 s

ninth case of Fig. 9, NeRFFE fails to edit the hair.

$$MSE_{mas} = MSE(M_{edi}, M_{res}) \tag{13}$$

To show the benefits of local control function, we calculate MSE_{OE} and MSE_{IE} by Formula 14. We use I'_{ori}, I'_{inv} and I'_{edi} respectively to represent the non-editing regions of I_{ori}, I_{inv} and I_{edi}. The results are shown in Table III . LC-NeRF best maintains the invariance of non-editing areas. It needs to be emphasized that NeRFFaceEditing uses VGG loss to explicitly constrain the image invariance of non-editing regions in local editing optimization. However, our method can achieve the best results without such explicit constraints when editing local regions.

$$\begin{aligned} MSE_{OE} &= MSE(I'_{ori}, I'_{edi}) \\ MSE_{IE} &= MSE(I'_{inv}, I'_{edi}) \end{aligned} \tag{14}$$

To evaluate facial identity, following [10], we calculate the quantitative metric multi-view facial identity consistency (ID), which is the mean Arcface [31] cosine similarity score between pairs of the same synthetic face rendered from two random camera poses. The ID of LC-NeRF is 0.65, which is higher than FENeRF [13] (0.61) and StyleNeRF [12] (0.62).

2) Local Texture Editing: Local texture editing allows users to modify the texture of an local region, which emphasizes naturality and harmony. FENeRF and NeRFFaceEditing are designed for local geometry editing and global texture editing, and do not support local texture editing. Thus we compare the local texture editing results of our method with IDE-3D. IDE-3D achieves local texture editing through extracting features from the two triplane features and combining them according to a mask to generate a new face image. This approach makes the generated images unnatural and there is a sense of splicing between different regions. LC-NeRF can directly change the texture latent code w_t of the certain local region and then fuse the edited high-dimensional texture features, so that the generated image is more natural and controllable. Due to the unavailability of the source code for local texture editing in IDE-3D, we invert the texture editing images given in their papers, which are not real human face images, but images generated by IDE-3D. Local texture editing results are shown in Fig. 7. It can be seen that IDE-3D hair texture editing results have a strong sense of border and contain jagged parts. In addition, after editing the mouth texture with IDE-3D, the geometry of the mouth is changed. Some of the results contain closed mouths, while others show open mouths. This shows that IDE-3D does not achieve effective decoupling of geometry and texture. At the same time, the edited mouth texture is unnatural and foggy, even spreading to non-mouth regions. The images edited by LC-NeRF are more natural and controllable, benefiting from our proposed local generators and high-dimensional feature fusion mechanism.

F. Efficiency

In order to demonstrate the efficiency and computational cost, we present the respective time required by each method to generate a single image. The results are shown in Table IV. All measurements are performed on an AMD EPYC 7302 CPU and a single GeForce RTX 3090 GPU.

G. Perception Study

We conduct a perception study to further verify the effectiveness of our method. We show the input image, the corresponding source mask and the edited mask, and

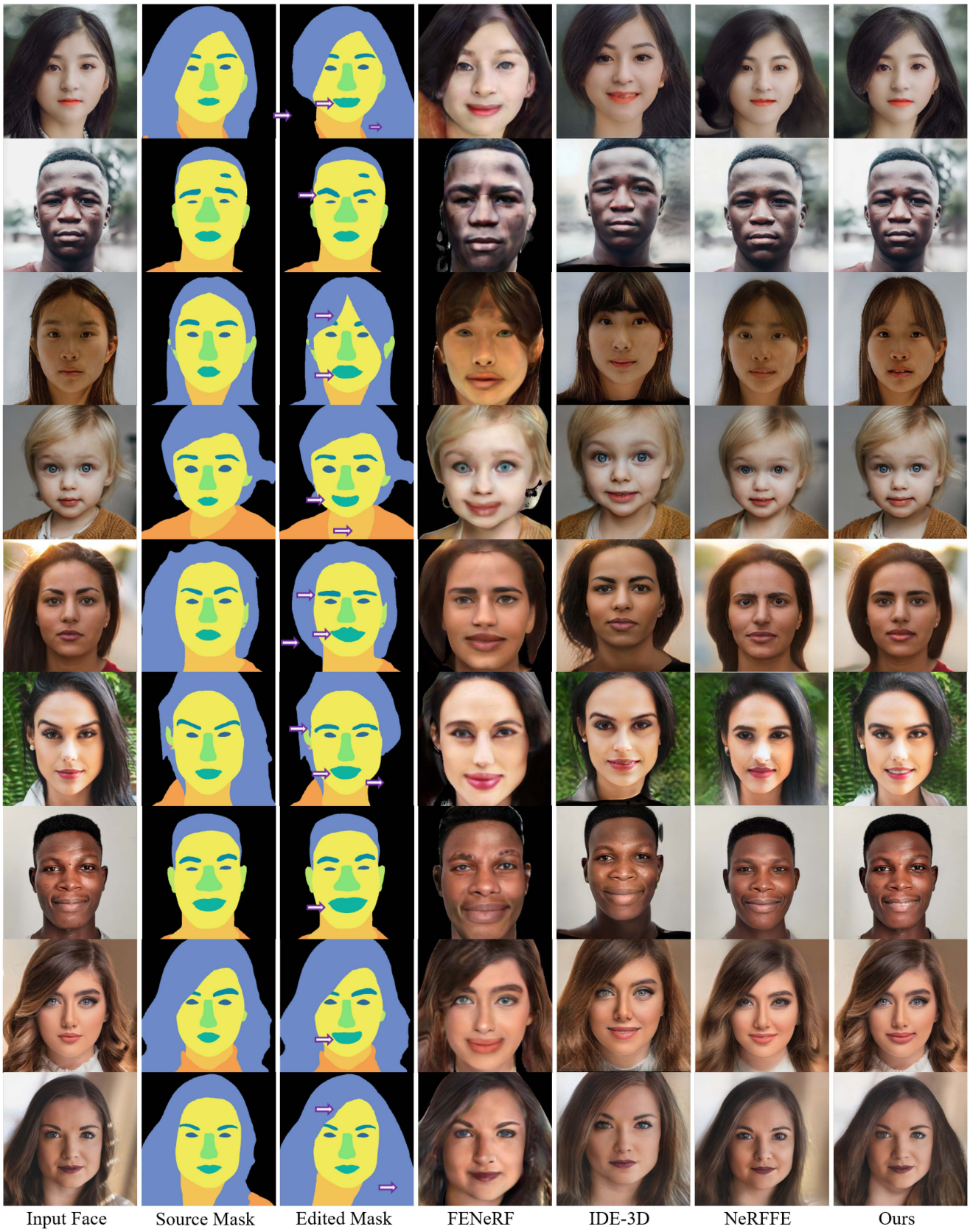

Fig. 9. In each case, we show the input face, the source mask, the edited mask (the arrow marks the editing region), and the face editing results of the four methods.

TABLE V
RESULTS OF PERCEPTION STUDY

Method	FENeRF	IDE-3D	NeRFFE	Ours
Image quality (↑)	1.12	2.46	2.93	**3.49**
Face consistency (↑)	1.13	2.34	3.00	**3.51**
Editing effect (↑)	1.72	2.44	2.58	**3.27**

four edited results (including FENeRF [13], IDE-3D [14], NeRFFaceEditing (NeRFFE) [15], and Ours) for each editing sample. Results from all four methods are placed in random order. In this perception study, each participant needs to rank images on 17 examples, some of which are shown in Figs. 6 and 9 in the following three aspects respectively:

- *Image quality*, which measures the quality of face images generated by different methods;
- *Face consistency*, which measures whether the edited face is consistent with the input face, that is, whether the face identity changes before and after editing;
- *Editing effect*, which measures the correctness and the quality of editing results.

In total, 40 participants participated in our perception study, and we got 40(participants)$\times$ 17(questions) $=$ 680 subjective evluation for each method. On average, each researcher spent 21.85 minutes on our survey. When computing the final score, the method ranked as first in each evaluation result translates to a score of four, the second translates to a score of three, the third a score of two, and the last a score of one.

As shown in Table V, our method achieves scores of 3.49, 3.51, and 3.27 in image quality, face consistency, and editing results respectively, exceeding the scores of FENeRF, IDE-3D, and NeRFFE. The visualization results are shown in the Fig. 8 in the form of a boxplot.

We also perform the ANOVA tests on the three aspects and get the F vlaues for image quality ($F = 252.39, p < 0.001$), face consistency ($F = 215.42, p < 0.001$), and editing effect ($F = 90.95, p < 0.001$). It can be clearly seen that our method has a significant improvement over the other three methods.

H. Ablation Study

The proposed SAFM uses semantic confidence to fuse local geometric features to get the global geometric feature, and then passes through the FC layer to get the density. To verify the effectiveness of SAFM, we replace it with two other architectural designs: each geometric sub-generator predicts the density separately, and then the final density is calculated by a) *Baseline1:* summing the densities; b) *Baseline2:* summing the densities weighted by semantic confidence s_i. Fig. 10 shows that the fusion operation will fail without the proposed *SAFM*. This demonstrates that the approach of predicting the total density through the global geometry feature is better than that of each sub-generator to predict its own density, resulting in a more spatially continuous density. This verifies the effectiveness of SAFM in semantic 3D-aware generation model.

Fig. 10. Ablation studies of the proposed *SAFM*.

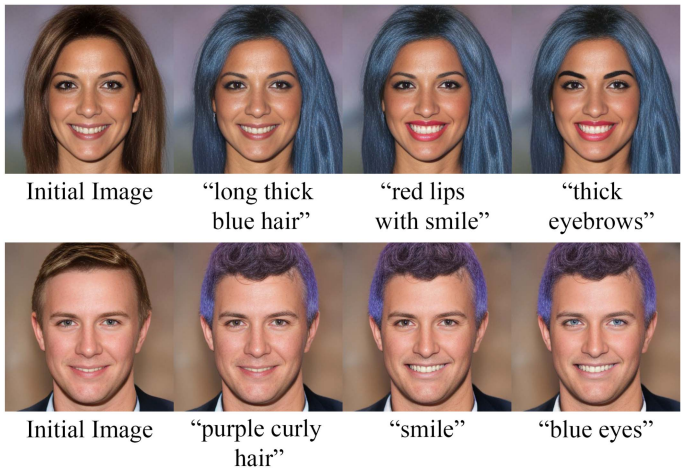

Fig. 11. Results of text-driven face editing. Give an initial image (left), LC-NeRF can edit it directly through the text. The figure shows the results of multiple local region edits, accumulative from left to right.

I. Text-Driven Face Editing

Text-driven face editing allows users to edit face directly using text, which is an effective and convenient way of editing. Therefore, we also explore applicaiton of LC-NeRF in text-driven image editing. We used StyleCLIP [5] with ViT-B/32 pretrained model for text guided latent manipulation. The driving text can directly optimize the $\mathcal{W}^+$ space latent code. In our experiments, generated images are controlled by short text clips, such as "thick eyebrows" and "red lips with smile", using CLIP loss [5]. We present sample edited images with corresponding prompt texts, with 100-300 latent optimizaiton steps, in Fig. 11. In each line of Fig. 11, editing results are accumulated, with each image using the optmized latent from the previous image as a starting point. The result shows that LC-NeRF allows for fine-grained control of facial features and accurate editing driven by text, enabling text-based editing of one facial feature without affecting other regions.

V. CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

We propose LC-NeRF, a local controllable and editable face generation method, which can generate view-consistent face images and semantic masks. Compared with the previous state-of-the-art face editing methods, LC-NeRF has achieved more fine-grained feature decoupling, including local region decoupling and decoupling of geometry and texture. Our method

achieves the best performance in face editing, which ensures the stability of non-editing regions and the consistency of face identities. Our method supports local mask editing, local and global texture editing, and can easily be extended to downstream tasks, such as text editing.

A limitation of our study is that while we can decouple the local regions and their geometry and texture, we are unable to finely control the local internal texture, including intricate details such as hair texture and facial wrinkles. Moving forward, our future research will focus on developing methods to achieve finer-grained control over the content of local texture.

Following StyleSDF [11], we choose SDF as our 3D representation. Each spatial point is relatively independent, which is more conducive to spatial decoupling. In the future, we will explore our performance on the latest 3D GAN, such as EG3D [10]. The proposed SAFM can be easily transplanted to any 3D GAN model.

REFERENCES

[1] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.

[3] T. Karras et al., "Alias-free generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 852–863.

[4] S.-Y. Chen et al., "DeepFaceEditing: Deep face generation and editing with disentangled geometry and appearance control," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 90:1–90:15, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459760

[5] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "StyleCLIP: Text-driven manipulation of StyleGAN imagery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2085–2094.

[6] Y. Shi, X. Yang, Y. Wan, and X. Shen, "Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 254–11 264.

[7] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, "SEAN: Image synthesis with semantic region-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5104–5113.

[8] S. Gu, J. Bao, H. Yang, D. Chen, F. Wen, and L. Yuan, "Mask-guided portrait editing with conditional gans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3436–3445.

[9] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. Eur. Conf. Comput. Vis.*, vol. 65, pp. 99–106, 2021.

[10] E. R. Chan et al., "Efficient geometry-aware 3D generative adversarial networks," 2021. [Online]. Available: https://arxiv.org/abs/2112.07945

[11] R. Or-El, X. Luo, M. Shan, E. Shechtman, J. J. Park, and I. Kemelmacher-Shlizerman, "StyleSDF: High-resolution 3d-consistent image and geometry generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 503–13 513.

[12] J. Gu, L. Liu, P. Wang, and C. Theobalt, "StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis," 2021, *arXiv:2110.08985*.

[13] J. Sun et al., "FENeRF: Face editing in neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7672–7682.

[14] J. Sun, X. Wang, Y. Shi, L. Wang, J. Wang, and Y. Liu, "IDE-3D: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis," 2022, *arXiv:2205.15517*.

[15] K. Jiang, S.-Y. Chen, F.-L. Liu, H. Fu, and G. Lin, "NeRFFaceediting: Disentangled face editing in neural radiance fields," in *Proc. ACM SIGGRAPH Asia Conf.*, 2022, pp. 1–9. [Online]. Available: https://doi.org/10.1145/3550469.3555377

[16] M. Niemeyer and A. Geiger, "GIRAFFE: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11448–11459.

[17] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680.

[18] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," 2020, *arXiv: 2002.10099*.

[19] E. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "Pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis," 2020, *arXiv: 2012.00926*.

[20] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7462–7473.

[21] Y. Deng, J. Yang, J. Xiang, and X. Tong, "GRAM: Generative radiance manifolds for 3D-aware image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10663–10673.

[22] T. Portenier, Q. Hu, A. Szabó, S. A. Bigdeli, P. Favaro, and M. Zwicker, "FaceShop: Deep sketch-based face image editing," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 99:1–99:13, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201393

[23] A. Chen, R. Liu, L. Xie, Z. Chen, H. Su, and J. Yu, "SofGAN: A portrait image generator with dynamic styling," *ACM Trans. Graph.*, vol. 41, no. 1, pp. 1–26, 2022.

[24] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5548–5557.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[26] S. Zhou, K. C. Chan, C. Li, and C. C. Loy, "Towards robust blind face restoration with codebook lookup transformer," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 30599–30611.

[27] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: A novel deep learning framework with meta-operators and unified graph execution," *Sci. China Inf. Sci.*, vol. 63, no. 12, pp. 222103:1–222103:21, 2020.

[28] D. Roich, R. Mokady, A. H. Bermano, and D. Cohen-Or, "Pivotal tuning for latent-based editing of real images," *ACM Trans. Graph.*, vol. 42, pp. 1–13, 2022.

[29] P. Zhu, R. Abdal, Y. Qin, J. Femiani, and P. Wonka, "Improved StyleGAN embedding: Where are the good latents?" 2020, *arXiv:2012.09036*.

[30] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, 2021.

[31] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong, "Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019. [Online]. Available: http://dx.doi.org/10.1109/cvprw.2019.00038

Wen-Yang Zhou is currently working toward the PhD degree with the Department of Computer Science and Technology, Tsinghua University. His research interests include computer graphics, 3D-aware generation, and computer vision.

Lu Yuan is working toward the undergraduation degree with Stanford University. Her research interests include computer graphics and computer vision.

Shu-Yu Chen received the PhD degree in computer science and technology from the University of Chinese Academy of Sciences. She is currently working as an assistant professor with the Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics.

Lin Gao (Member, IEEE) received the PhD degree in computer science from Tsinghua University. He is currently an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the AG young Researcher award. His research interests include computer graphics and geometric processing.

Shi-Min Hu (Senior Member, IEEE) received the PhD degree from Zhejiang University in 1996. He is currently a professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He has published more than 100 papers in journals and refereed conferences. He is the editor-in-chief of *Computational Visual Media*, and on editorial boards of several journals, including *Computer Aided Design and Computer & Graphics*. He is a senior member of ACM, and fellow of CCF and SMA.