StrokeFaceNeRF: Stroke-based Facial Appearance Editing in Neural Radiance Field

Xiaojuan Li[1,2] Dingxi Zhang[1] Shuyu Chen[2] Fenglin Liu[1,2]
[1]University of the Chinese Academy of Sciences
[2]Institute of Computing Technology, Chinese Academy of Sciences
{lixiaojuan,chenshuyu, liufenglin21s}@ict.ac.cn zhangdingxi20a@mails.ucas.ac.cn

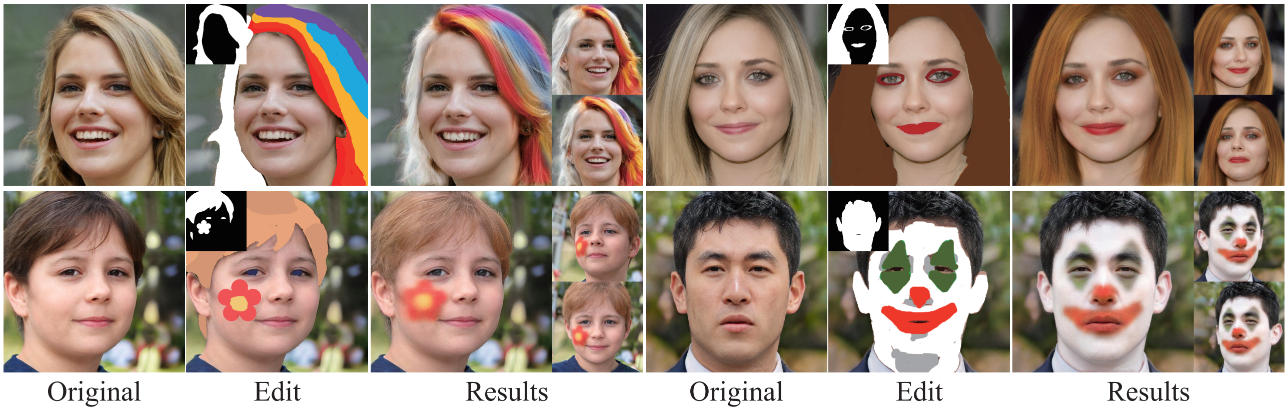


Figure 1. StrokeFaceNeRF enables versatile high-quality color stroke-based appearance editing for facial NeRFs. Users can draw strokes over a rendered facial image from any angle and define the editing region using masks. The third column showcases the photo-realistic free-view rendering results with exceptional color correspondence.

Abstract

Current 3D-aware facial NeRF generation approaches control the facial appearance by text, lighting conditions or reference images, limiting precise manipulation of local facial regions and interactivity. Color stroke, a user-friendly and effective tool to depict appearance, is challenging to edit 3D faces because of the lack of texture, coarse geometry representation and detailed editing operations. To solve the above problems, we introduce StrokeFaceNeRF, a novel stroke-based method for editing facial NeRF appearance. In order to infer the missing texture and 3D geometry information, 2D edited stroke maps are firstly encoded into the EG3D's latent space, followed by a transformer-based editing module to achieve effective appearance changes while preserving the original geometry in editing regions. Notably, we design a novel geometry loss function to ensure surface density remains consistent during training. To further enhance the local manipulation accuracy, we propose a stereo fusion approach which lifts the 2D mask (inferred from strokes or drawn by users) into 3D mask volume, allowing explicit blending of the original and edited faces. Extensive experiments validate that the proposed method outperforms existing 2D and 3D methods in both editing reality and geometry retention.

1. Introduction

Realistic 3D human face generation is a popular topic and has wide range applications in character design, virtual meeting and education. Traditional 3D facial modeling methods rely on software like Maya and Zbursh to construct mesh models while employing texture mapping for facial appearances. However, these methods are time-consuming and labor-intensive, especially when editing local appearance details and aiming for photorealistic results. Neural Radiance Field (NeRF) [24], a new 3D representation technique, easily render photo-realistic images with the input of multi-view images. The further combination of Generative Adversarial Networks (GAN) [11] with NeRF [6, 12, 25, 31] has enabled the generation of high-quality facial NeRF through random sampling. Nevertheless, precise control over appearance details remains a challenge.

Efforts to enhance controllability involve disentangling geometry and appearance during facial NeRF generation. Sun et al. introduced the FE-NeRF [33] that employs separate latent codes to control the intricacies of facial geometry and appearance. Subsequent advancements [17, 32] utilize decoupling techniques based on triplane representations, leading to the facial generation results with significantly enhanced quality. Despite their success in good geometry and appearance swapping, these methods heavily rely on random sampling or reference images for global appearance editing. This reliance poses challenges in the obtainment of desired reference images and precludes precise control over localized appearance attributes, such as selective hair color alteration.

Compared to reference images, colored strokes, functioning as an interactive tool, offer a precise depiction of local appearance details. In stroke-based image editing, one category of methods [18, 27] relies on conditional GAN and completion framework to edit local regions with the stroke guidance. However, artifacts may arise in the editing boundary and result in less realistic images for large-area editing operations, as shown in Figure 6. Another category of methods [23, 28] employs pre-trained facial generation models as decoders, mapping colored strokes to the latent space of the generation model to generate edited facial results. However, these methods struggle to preserve the unchanged geometric features of the input face during editing, as demonstrated in Figure 6. Importantly, these techniques are specifically tailored for 2D facial image editing and cannot be directly applied to 3D facial radiance fields.

This paper presents StrokeFaceNeRF, a framework for editing facial NeRF appearance based on colored strokes, enabling precise modifications of color and lighting in 3D faces. Leveraging 2D facial image generation techniques such as StyleGAN2 [22] and semantic segmentation methods [39], we construct a dataset of faces with identical geometry but varying appearances, generating corresponding colored stroke maps using bilateral filtering and semantic clustering algorithms. To achieve appearance editing, directly encoding edited stroke maps into the latent space of the pre-trained facial NeRF generation model [6] leads to texture smoothness and undesirable geometry distortion due to high-frequency information absence and user's inaccurate drawing of strokes. To solve this problem, we propose a transformer-based editing network, which fuses the original and stroke-encoded latent codes to modify appearance while preserving the original geometry in edited regions. Furthermore, we introduce a geometry loss term during training to ensure surface density remains unchanged, improving detail preservation. To further maintain unchanged features in non-edited areas for precise editing, we design a volumetric fusion algorithm that integrates localized regions before and after editing in 3D space, enabling localized appearance editing effects in facial models.

Our contributions are summarized as follows:

- We propose the first color stroke-based facial NeRF editing framework to generate free-view realistic appearance editing facial images.
- We design a novel geometry density loss to train the transformer-based latent editing network, which effectively preserves the geometry detail during facial appearance modifications.
- We introduce a volumetric fusion algorithm to precisely edit local regions while maintaining the 3D consistency in unedited regions.

2. Related Work

2.1. Facial Image Editing with Colored Strokes

Existing 2D facial image editing methods using colored strokes can be divided into two categories. One group of methods utilizes conditional Generative Adversarial Networks (cGANs) to accomplish stroke-guided image generation. Sangkloy et al. [30] utilize sketches and colored strokes to directly generate realistic images through a translation network. However, this method fails to achieve localized area editing. To solve this problem, FaceShop [27] utilizes image completion techniques and 2D mask marking for localized facial image editing based on sketches and colored strokes. Jo et al. [19] further introduced SC-FEGAN that employs gated convolutions and new style losses for training, enabling a broader range of editing operations. Xiao et al. [37] design a hair generation network with self-attention mechanisms, achieving more natural and realistic hair editing effects. However, these methods are sensitive to input conditions and result in smooth edits lacking texture information in edited regions. Additionally, they tend to produce artifacts at the editing boundary due to the image completion techniques.

Another category of methods utilizes pre-trained generative models to generate high quality faces instead of pixel-wise translation. Bau et al. [4] first utilize reverse optimization to achieve image shape and color editing in the GAN's latent space. pSp [28] further designs an image encoding network to map various types of input images, such as sketches and color strokes, into StyleGAN2's latent space to generate corresponding facial images. In order to generate highly editable latent codes, Tov et al. [35] further constrained the latent codes with a discriminator. However, these methods, as they represent the entire image with latent vectors, struggle to achieve localized editing effects. Instead of GAN dependence, Meng et al. [23] employed pre-trained diffusion models to achieve facial image editing based on colored stroke maps. Images with user-drawn strokes are added Gaussian noise, then progressively denoised to synthesize realistic facial images. However, it is

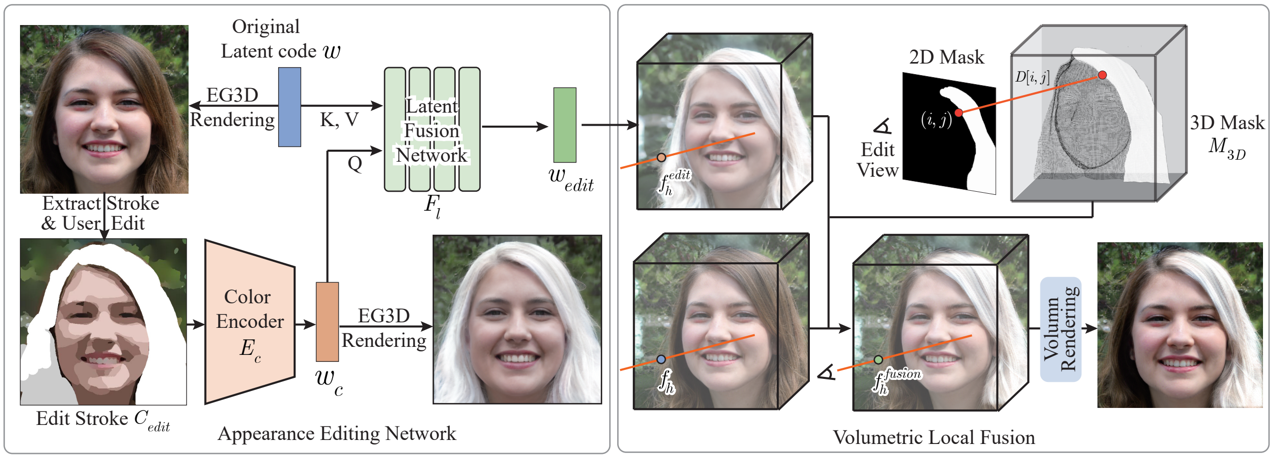


Figure 2. 3D Facial Appearance Editing Framework. Given the original latent code w, free-view images are rendered by EG3D. Users introduce editing operations, resulting in the creation of a color stroke image C_{edit}. This stroke map is encoded by Color Encoder E_c to generate w_c, which is fused with original latent code w by latent fusion network to generate w_{edit}. Utilizing an additional 2D mask, the volumetric fusion approach facilitates targeted appearance editing within specific regions.

challenging to directly apply this method to facial neural radiance field editing. Additionally, because of the direct input of coarse stroke maps, it is challenging for the above methods to maintain geometric features while performing appearance modification in editing regions.

2.2. Editing Facial NeRF

Combining generative adversarial networks with neural radiance fields enables the synthesis of realistic facial images from arbitrary viewpoints. Graf [31] first adds extra latent codes to NeRF's fully connected network to support face generation and use multiscale discriminators for supervision. Giraffe [25] designs a new framework that employs volume rendering to obtain feature maps and a convolutional framework to generate facial images. StyleSDF [26] and GRAM [8] utilize representation methods based on SDF and implicit planes, respectively, producing higher-quality and more 3D consistent results. EG3D [6] propose a facial NeRF generation framework based on triplane representation and StyleGAN2 [22] generator. However, these methods can only synthesize faces through random sampling and lack the ability for editing.

To generate controllable facial NeRF, Jo et al. [18] use sketches, black-and-white images, and text as control conditions. Gao et al. [10] propose a facial NeRF generation and editing framework based on sketches to enhance synthesis quality and achieve fine editing. However, this method can only edit facial geometry. Another line of work decouples facial NeRF's geometry and appearance using semantic segmentation. FE-NeRF [33] achieves geometric editing through reverse latent code optimization and appearance editing through random sampling. IDE-3D [32] constructs geometry triplanes and appearance triplanes, and designs a semantic map encoder to generate higher-quality facial results. NeRFFaceEditing [17] apply AdaIN [15] to triplanes features, enabling the disentanglement of geometry and appearance control. Although the above methods achieve effective appearance modification, reference images or latent codes are required instead of user's flexible interaction. More importantly, the above methods are designed for the global appearance editing and cannot be directly utilized for local editing.

3. Methodology

We propose a framework, StrokeFaceNeRF, enables users to utilize colored strokes for facial NeRF appearance editing. We employ the EG3D [6] generator architecture, a proficient 3D-aware Generative Adversarial Network (GAN) that combines the StyleGAN2 [22] architecture and neural rendering to produce high-quality 3D shapes. Given a latent code, a triplane representation can be obtained from a StyleGAN2 backbone. The neural renderer then aggregates features from these triplanes and predicts feature images based on a specified camera pose. Subsequently, a super-resolution module is applied to upsample and refine these raw neurally rendered images, resulting in high-quality, multiview-consistent images. More details of the generator can be found in [6].

Our method comprises three key components: dataset construction, colored stroke editing network, and 3D local fusion module. Initially, we introduce a dataset construction approach to acquire facial images with consistent geometry but varying appearances, along with their corresponding colored stroke images as detailed in Section 3.1. Subsequently, an appearance editing network, described in Section 3.2, is established within the latent space of EG3D to predict the 3D faces. Additionally, in Section 3.3, we propose a 3D local fusion method to enable effective appearance editing while meticulously preserving features in non-edited regions.

3.1. Dataset Construction

To supervise the training of network models, the training data needs to simulate the process of facial appearance editing based on colored strokes. Thus, the training tuples include input faces, edited faces, and corresponding colored stroke maps. Dataset construction involves three steps as shown in Figure 3: (1) building a collection of facial latent codes in EG3D's latent space, obtained through random sampling and real images inversion; (2) Constructing facial tuples with identical geometry but different appearances. Two different methods were employed: one based on semantic segmentation and the other based on StyleGAN2 generative networks; (3) generating colored stroke maps using bilateral filtering and semantic clustering methods.

We utilize two approaches to synthesize paired images with the same geometry but different appearance. *Semantic Segmentation-Based Augmentation.* Given the rendered facial images, we utilize BiSeNet [39] to get semantic segmentation masks. Then, for each local semantic region, we represent it into HSV space, where the Hue channel is uniformly replaced with random color, while the Saturation and Value channels are added random perturbation. Finally, these regions are merged into original images to imitate appearance editing. *StylegGAN2-Based Augmentation.* We further utilize pre-trained generator to synthesize more diverse and realistic training images. Given the rendered images, ReStyle [3] encodes them into StyleGAN2 latent space, followed by style mixing to change the facial appearance while preserving the same geometry. More details can be found in Supplementary.

With the paired appearance modification images, corresponding color strokes are required to imitate real editing situation. Typically, users drawn strokes are simple and monochromatic. Therefore, synthesized color stroke images must preserve the original image's color and geometric layout while eliminating high-frequency texture features and local details. To achieve this objective, we employ the bilateral filtering method [34] to effectively preserves the edge features of the facial image while eliminating high-frequency details. Specially, the initial image undergoes median filtering, followed by iterative application of bilateral filtering, as explained in the Supplementary. Nevertheless, some regions still exhibit smooth transitions between different colored areas after filtering, posing challenges for users aiming to replicate specific effects. To tackle this issue, we employ a pixel value clustering technique based on semantic information. Initially, the facial image undergoes semantic segmentation. Within the hair area, clustering allows all pixel colors to be replaced by the nearest cluster center color, thereby eliminating smooth transition regions.

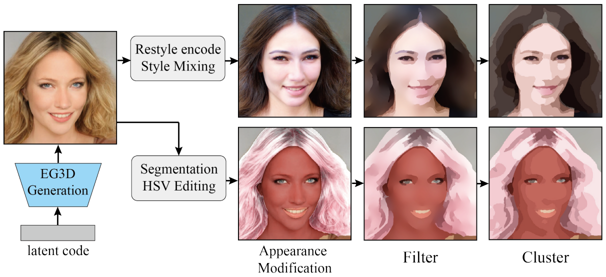


Figure 3. Dataset construction pipeline. Given EG3D rendering images, the appearance is changed by two different approaches. Then, the color stroke images are synthesized by image filter and cluster.

The same methodology is applied to the facial area, yielding the final color stroke image.

3.2. Appearance Editing Network

In our framework, the target face NeRF to be edited is represented as a latent code w within EG3D's latent space. The latent code can be obtained either by random sampling in EG3D or by applying PTI [29] to invert real images. Utilizing StyleGAN generator and volume rendering, free view facial images can be generated, followed by the approaches in Section 3.1 to extract color stroke images. Users then modify these color stroke images to obtain the edited color stroke image C_{edit}. As illustrated in Figure 2, the input to the appearance editing network consists of the original latent variable w, the edited color stroke image C_{edit}, and the output is the edited latent variable w_{edit}.

The appearance editing network comprises two main components: the **Color Encoder** E_c and the **Latent Fusion Network** F_l. Specifically, given the edited color stroke image C_{edit}, the color encoder that has a similar structure to [28], predicts the latent code w_c. While this latent code generates facial results with color editing effects, it fails to preserve the original facial features, such as the geometric structure of the hair and the original skin color in non-edited areas, as shown in Figure 2. Therefore, a latent code fusion network is designed to utilize the information from the original latent variable w, thus better preserving the original facial features. The latent fusion network F_l adopts a cross-attention transformer block to predict the fused latent variable w_{edit}. This design implicitly extracts the color information from w_c and geometry information from w, guaranteeing a thorough integration of both color-edited and original facial features. In this configuration, the latent code w_c is employed as query tokens, whereas the original latent code w serves as key and value tokens. We also normalize over queries instead of keys as described in [14] to ensure a continuous refinement of w_c during the fusion process with value tokens while mitigating the potential influence of w being too large.

$$w_{\text{edit}} = F_l(w, E_c(C_{\text{edit}})) \tag{1}$$

3.3. Volumetric Local Fusion

While direct rendering by w_{edit} yields good editing effects within targeted editing areas, as depicted in Figure 7, noticeable undesirable discrepancies emerge in unedited regions. Moreover, intricate editing tasks, such as selectively dyeing specific portions of hair or drawing specific patterns on faces, pose challenges for the latent code-based representation because of its overarching global control inherited from StyleGAN2 backbone. To address these issues, we propose a volumetric fusion approach to enable more refined editing operations.

In order to facilitate local editing, our method introduces support for an additional 2D mask input that indicates the editing region. This 2D mask is further lifted into 3D mask volume to support volumetric fusion. Specifically, we propose a depth guidance approach to sample a set of points near the facial surface. In the volume rendering, for each ray $r_{i,j}$ of the pixel at the ith row and jth column, the camera's position and ray direction are denoted as $o_{i,j}$ and $d_{i,j}$, respectively. After rendering the original faces, we get the corresponding depth value $D[i, j]$. Then, for each ray, we uniformly sample K points with offset $\triangle x_k$ within range $[-0.1, 0.1]$. The sampled 3D points set S for the original face is denoted as:

$$S = \cup_{i,j} \cup_{k=1}^{K} (o_{i,j} + (D[i,j] + \triangle x_k) \cdot d_{i,j}) \qquad (2)$$

where the default value of K is 5. Notably, as illustrated in Figure 2, we selectively employ the rays corresponding to pixels within the 2D mask regions.

Subsequently, this generated 3D point set undergoes a transformation into a 3D volumetric mask. Specifically, we initialize a $256 \times 256 \times 256$ 3D volume with zero values throughout the space. The coordinates of the 3D points, along with their neighboring positions, are designated as 1, indicating 3D editing regions. To ensure smoother boundaries, we utilize a three-dimensional dilation and Gaussian filtering, yielding the 3D mask denoted as M_{3D}. Additionally, We implement a mask padding strategy, which introduces direction offsets to the camera poses and extends the mask boundary. This prevents incomplete editing artifacts during view changes. Further details regarding the padding technique are available in the Supplemental material.

To render realistic editing images, the original latent variable w and the edited latent variable w_{edit} are used to synthesize the original and edited triplane features, respectively. For the sampling points h in space during volume rendering, the corresponding original feature f_h and edited feature $f_{\text{edit}}(h)$ are generated. To achieve the editing fusion effect, the features are fused based on the 3D mask M_{3D}:

$$f_{\text{fusion}}(h) = f_{\text{edit}}(h) \cdot M_{\text{3D}}[h] + f(h) \cdot (1 - M_{\text{3D}}[h]) \quad (3)$$

The fused features f_{fusion} are passed through the decoder of EG3D to obtain color and density information. Afterward, they are processed through the volume rendering and upsampling modules to generate the final edited faces.

3.4. Training Strategy

During the training of Appearance Editing Network, the geometry details should be preserved while the appearance editing effectiveness should not be influenced. This is nontrivial because the 2D supervision lacks detailed 3D structure constrain and the synthesized training data has minor geometry distortion because of Style-mixing's limitation. To solve this problem, we train our network with the follow strategy.

Geometry Density Loss. Based on our NeRF representation, we design a novel density loss to enforce the 3D geometry consistency between the original and edited faces. Specifically, leveraging the 3D points generation approach described in Equation 2, a collection of 3D points is generated. Notably, we use all the rays during the rendering instead of rays in specific regions. These 3D sample points are further projected onto triplane space to query corresponding triplane features that are subsequently decoded into color and density through volume rendering. We minimize the density value between the original and edited faces to effectively preserve the facial geometry while decoupling the appearance:

$$L_{\text{density}} = \sum_{x \in S, x' \in S'} ||\psi_{\text{density}}(\mathcal{F}(x)) - \psi_{\text{density}}(\mathcal{F}'(x'))||_1 \qquad (4)$$

where $\mathcal{F}$ denotes the triplane query process and ψ_{density} denotes density decoder. x' and S' are the 3D points and set for edited faces. These points are sampled with the same offset but new depth map D_{edit} of edited faces. Edited triplane features $\mathcal{F}'$ synthesized by w_{edit} are used for density calculation.

We also utilizes the same loss functions as the pSp framework [28] to match the original stroke distribution. The loss function includes the per-pixel L_2 loss between the ground truth image I_{GT} and the edited image generated by EG3D G:

$$L_{img} = ||I_{GT} - G(w_{\text{edit}})||_2 \qquad (5)$$

Here, $||.||_2$ denotes the Euclidean distance. To better constrain the reconstruction quality, a perceptual distance loss term is further employed:

$$L_{LPIPS} = ||F(I_{GT}) - F(G(w_{\text{edit}}))||_2 \qquad (6)$$

Where F represents an image feature extraction method based on VGG [20]. In order to better constrain the images generated to preserve the identity features of the original faces, an identity constraint term is employed:

$$L_{ID} = 1 - \langle R(I_{GT}, R(G(w_{\text{edit}}))\rangle \qquad (7)$$

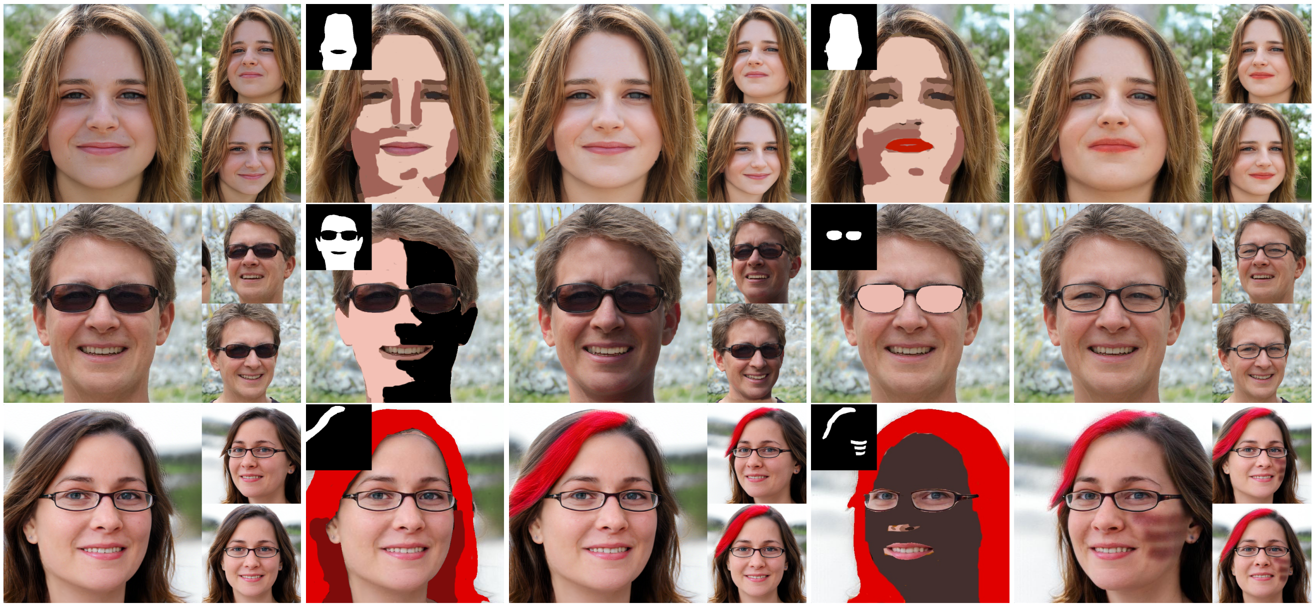


Figure 4. Color stroke-based facial NeRF appearance editing results. Given the original faces (a), users drawn the editing color strokes (b)(d), our methods generates the corresponding editing results (c)(e). Additional masks are shown in top left corner, which is the same as stroke's shape in the first two rows, and drawn by users in the last row.

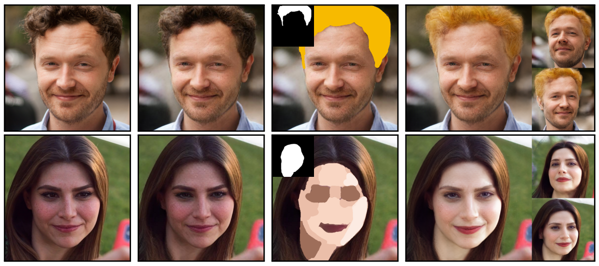


Figure 5. Real image appearance editing results.

R represents the pre-trained facial recognition network, ArcFace [7], and $\langle . \rangle$ denotes cosine similarity. Regularization constraints are added to ensure that the edited latent variables are as close as possible to the mean latent variables $\overline{w}$, making the generated faces more natural.

$$L_{reg} = ||w_{\text{edit}} - \overline{w}||_2 \tag{8}$$

The overall loss function is given by:

$$L = \alpha_1 L_{\text{density}} + \alpha_2 L_{\text{edit}} + \alpha_3 L_{LPIPS} + \alpha_4 L_{ID} + \alpha_5 L_{reg} \tag{9}$$

Here, α_1, α_2, α_3, α_4 and α_5 represent the weighting coefficients for each loss term, with values of 0.05, 2.0, 2.0, 0.2, and 0.005, respectively. In the first stage training, we set $\alpha_1 = 0$ because only appearance is expected to be encoded in this stage.

4. Experiments

Dataset: Our training dataset comprises original faces generated by EG3D random sampling and real images inversion [1] of CelebA-HQ [21], amounting to 47,134 examples. Appearance-edited faces and corresponding color stroke images are constructed using the method detailed in Section 3.1, resulting in 94,268 paired samples in total.

Training details: We use a single NVIDIA GeForce RTX 3090 GPU to train our framework. The batch size is 4, with the Ranger optimizer [38] and a learning rate of 1e-4. Each training stage involved 30k iterations, completing the final network training process.

Evaluation metrics: We use the following common metrics: Fréchet Inception Distance (FID) [13], Kernel Inception Distance (KID) [5] and Identity Consistency (ID) from [16]. To evaluate the geometry consistency in both 2D and 3D methods, we computed the Structural Similarity Index (SSIM) [36] metric for the edges obtained using the Sobel [9] operator in the edited regions of the generated images compared to the real images, denoted as E-SSIM. A higher E-SSIM value in the edited regions indicates a better preservation of the geometry in those areas.

4.1. Results

Figure 4 illustrates the 3D facial NeRF appearance editing based on color strokes. Benefit from the 3D representation in our framework, users can select arbitrary view point to render images. Then, users could draw color strokes, which are overlaid on extracted strokes extracted based on methods described in Section 3.1 to serve as the edited stroke image. Our framework takes an additional 2D mask inputs, which either have the same shape with drawing strokes, or

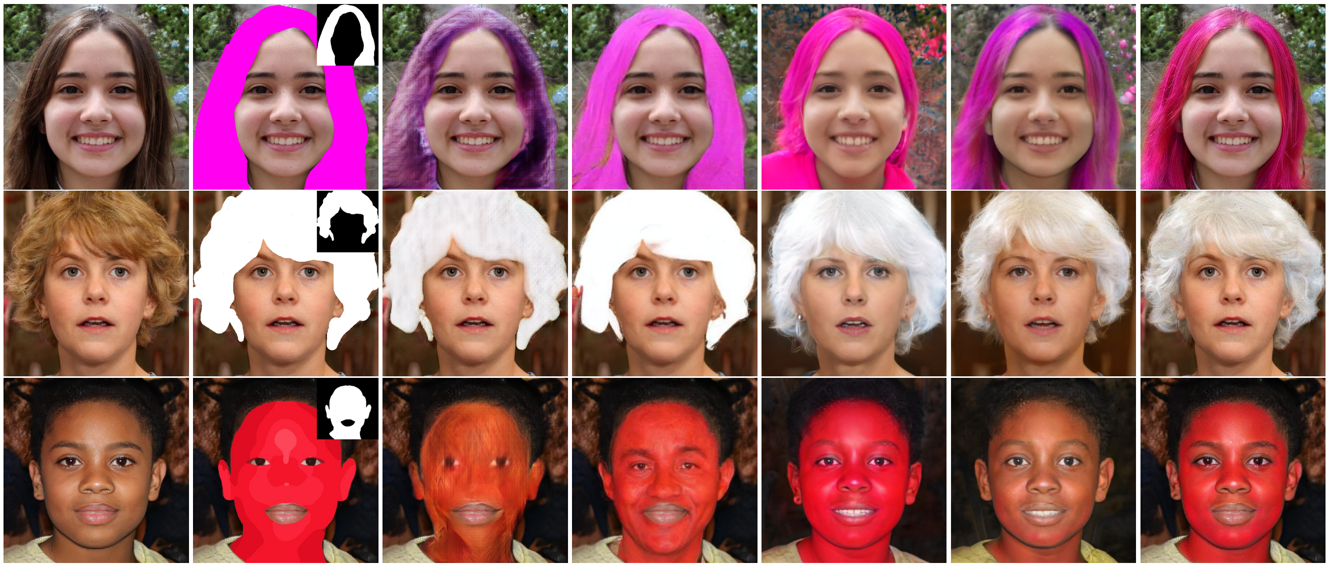


Figure 6. Comparison with existing color stroke-based facial editing methods. SC-FEAGN[19] and SDEdit[23] are 2D image editing approaches that generate texture smoothness. pSp[28] and pSp-mixing globally changes the 3D faces. Our method generates the best results on realism and identity preservation.

additionally drawn by users. As shown in Figure 4(c)(e), our method allows the precise control over the skin, hair, and lip colors. It is noteworthy that the geometric details are well preserved as the original faces during appearance editing, as demonstrated by the wrinkle details in the 1st row and hair structure in the 3rd row. Unedited regions, such as hair (1st row) and background, also retain the same as the original faces. The 3rd row of Figure 4 showcases more complex local appearance editing results. To achieve diverse editing effects, users draw finer masks (c) to mark local areas, such as a cluster of hair or specific facial patterns. The 3D local fusion algorithm enables editing in marked areas, resulting in more complex and diverse editing effects.

As shown in Figure 5, given a real facial image, the original facial image is reconstructed in latent space using the PTI algorithm [29] (b). Similarly, users draw color stroke images (c) and edit area masks (d). Our method generates corresponding facial NeRF appearance editing results and obtains multi-view facial outcomes.

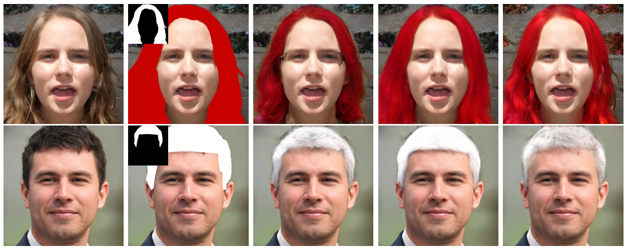
4.2. Comparisons

Table 1. Quantitative evaluation using FID, KID $\times 10^2$, mean and std value of identity consistency (ID), E-SSIM $\times 10^2$ on baseline models and our methods.

	FID (↓)	KID(↓)	ID(↑)	E-SSIM(↑)
SC-FEGAN [19]	47.18	3.41	0.50 ± 0.37	74.36
SDEdit [23]	37.61	2.52	0.41±0.41	74.52
pSp [28]	40.39	2.49	0.66 ± 0.11	78.94
pSp-mixing	31.32	2.00	0.73± 0.09	81.50
Ours	**16.86**	**0.83**	**0.85± 0.10**	**84.30**

Since there are no prior works on 3D-aware stroke-based image synthesis, we retrained the pSp [28] model on EG3D using our dataset. To ensure fair comparisons, we further applied style-mixing on pSp's predicted latent code by replacing 7-14 layers with original latent codes. We also compare with 2D image editing methods, including SC-FEGAN [19] and SDEdit [23]. Officially released codebases were used for all comparisons. As shown in Figure 6, given the original faces(a), color strokes and masks(b), since the editing areas are too large to complete, SC-FEGAN [19] generates less realistic results(c), with facial details appearing blurry in larger editing areas. Based on diffusion prior knowledge, SDEdit [23] generates more realistic results(d). However, the texture is still too smooth due to the geometry information absence of color stokes. Different from 2D image completion frameworks, pSp [28] predicts a latent code, which, however, totally modify the geometry compared with original faces. Unedited regions, such as background, are also changed. Even utilizing the style-mixing approach (f), there are also undesirable changes and the degree of editing has been largely affected. Compared with existing approaches, our method generates more realistic faces, which shows good faithfulness to the color strokes and precisely preserve the geometry details, even the sophisticated curly hair structure is well maintained.

Quantitative Comparison. To further validate the superiority of our method, we conduct a quantitative evaluation. We generated 1000 facial editing examples in FFHQ datasets, following the data generation method outlined in Section 3.1. We calculated the relevant metrics between all algorithm-generated results and real images. As shown in



(a) Original (b) Input (c) w/o L_{den} (d) depth loss (e) Ours

Figure 7. The results of ablation study for density loss. w/o L_{den} method is trained without geometry density loss while depth loss method is trained with an alternative depth image loss.

Table 1, our method has achieved the best results on all metrics, which proves that our method generates more realistic faces (FID, KID). The original identity feature (ID) and geometry details (E-SSIM) are also better preserved.

4.3. Ablation Study

Table 2. Ablation study comparison. All models are trained on the datasets described in Section 3.1 and evaluated on 1000 randomly sampled EG3D data. Metric calculations follow the methods outlined in Table 1.

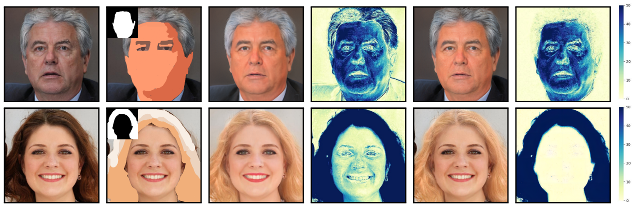
	FID (↓)	KID(↓)	ID(↑)	E-SSIM(↑)
w/o density loss	21.44	1.07	0.67 ± 0.22	77.35
w/ depth loss	23.14	1.23	0.75 ± 0.17	80.10
w/o mask fusion	28.33	1.72	0.82± 0.07	82.97
Ours	**16.86**	**0.83**	**0.85± 0.10**	**84.30**

We conduct ablation experiments on EG3D random sampled data to justify the necessity of each component. To validate the effectiveness of the proposed geometry density loss, we presents the results of training without density loss L_{den} and with an alternative depth image loss. Specifically, we employed MSE Loss between the original depth image and the generated one to constrain the network. The results in Figure 7 (d)(e) and Table 2 demonstrate the superiority of our density loss in preserving facial geometry features and maintaining identity consistency.

Figure 8 indicates our volumetric local fusion module' effectiveness in maintaining consistency for non-edited areas between and the original and edited face. As shown in Figure 8 (d), without using 3D local fusion, non-edited areas undergo significant changes, such as brightening hair in the first row and reddening the face in the second row. In contrast, our method (Figure 8 (e)) perfectly maintains consistency in non-edited areas without affecting the editing effects in edited regions. Figures 8 (d) and (f) display computed heatmap differences between pre-edited and post-edited images.

5. Conclusion

In this study, we propose a facial NeRF appearance editing method based on color strokes, enabling the modification of facial appearance features while preserving the original facial geometry. Initially, a data construction approach is introduced, leveraging StyleGAN2 and semantic segmentation networks to create paired data with consistent geometry and diverse appearances, along with corresponding color strokes. Subsequently, an appearance editing network is constructed within the pre-trained generative network's latent space. The network employs a color stroke encoder to extract color features from input stroke images and utilizes a latent variable fusion network to achieve appearance editing while preserving geometric features. To maintain unchanged regions, a 3D local fusion method is introduced, enabling effective editing of localized appearance while preserving the facial identity features. Experimental results demonstrate the superiority of our method in generating higher-quality digital facial results compared to existing approaches.



(a) Original (b) Input (c) w/o fusion (d) Heatmap 1 (e) Ours (f) Heatmap 2

Figure 8. The results of ablation study for local region fusion. w/o fusion are the results without 3D local fusion method. Corresponding heat map indicates the difference between the edited images and the original images.

Despite achieving high-quality color editing results through color strokes, our method faces several challenges. Firstly, for highly detailed areas such as eyeballs, our method struggles to yield effective editing results. Handling these regions separately through segmentation and dedicated data processing could enhance editing outcomes. Additionally, due to limitations in the training dataset, our method encounters difficulties in handling overly complex color strokes, such as rainbow-colored hair, tending to average color strokes instead. To address this, generating distinct colors separately, followed by mask creation and blending, could yield superior results. Future research endeavors could extend the proposed algorithm from 3D digital faces to other types of 3D models, achieving more generalized editing effects. Moreover, combining color strokes with line drawings and semantic segmentation maps could enable simultaneous editing of 3D digital facial geometry and appearance.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4432–4441, 2019. 6

[2] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019. 11

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 4

[4] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020. 2

[5] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 1, 2, 3, 12

[7] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 6

[8] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[9] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*. Wiley New York, 1973. 6

[10] Lin Gao, Feng-Lin Liu, Shu-Yu Chen, Kaiwen Jiang, Chun-Peng Li, Yu-Kun Lai, and Hongbo Fu. Sketchfacenerf: Sketch-based facial generation and editing in neural radiance fields. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2023)*, 42(4):159:1–159:17, 2023. 3

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[12] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022. 1

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[14] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11337–11346, 2022. 4, 13

[15] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1510–1519. IEEE Computer Society, 2017. 3

[16] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, and Feiyue Huang Jilin Li. Curricularface: Adaptive curriculum learning loss for deep face recognition. pages 1–8, 2020. 6

[17] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Lin Gao. Nerffaceediting: Disentangled face editing in neural radiance fields. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 3

[18] Kyungmin Jo, Gyumin Shim, Sanghun Jung, Soyoung Yang, and Jaegul Choo. Cg-nerf: Conditional generative neural radiance fields for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 724–733, 2023. 2, 3

[19] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1745–1753, 2019. 2, 7, 13

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 5

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 2, 3, 11

[23] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2, 7, 14

[24] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1

[25] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 3

[26] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 3

[27] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972*, 2018. 2

[28] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 2, 4, 5, 7, 12, 14

[29] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 4, 7, 11

[30] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2017. 2

[31] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1, 3

[32] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *ACM Transactions on Graphics (ToG)*, 41(6):1–10, 2022. 2, 3

[33] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 2, 3

[34] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998. 4, 11

[35] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4): 1–14, 2021. 2

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[37] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: Deep sketch-based hair image synthesis. *arXiv preprint arXiv:2109.07874*, 2021. 2

[38] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 635–652. Springer, 2020. 6

[39] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129: 3051–3068, 2021. 2, 4

StrokeFaceNeRF: Stroke-based Facial Appearance Editing in Neural Radiance Field

Supplementary Material

In this supplement, we provide a comprehensive overview of the color stroke image generation methods (Section 6) and additional visual results (Section 7). We also introduce our user interface in Section 8. Subsequently, we delve into the specifics of our implementation (Section 9), including network architecture details and mask padding strategy. Further insights into our experiments are discussed in Section 10, encompassing additional ablation studies on the two-stage training and further explanations for experiments. Additionally, we address potential artifacts (Section 11) that serve as subjects for future research. We encourage readers to explore the accompanying supplemental video, which features live demonstrations of our appearance editing process alongside additional visual results.

6. Color Stroke Images Generation

During the dataset construction, we synthesize facial images and corresponding stroke maps. Typically, users draw simple monochromatic strokes to simplify interaction. Therefore, synthesized color stroke images should preserve the original image's color and geometric layout while eliminating high-frequency texture features and local details.

Directly applying methods like Gaussian filtering to remove high-frequency information often results in edge blurring, leading to the generation of unreasonable color stroke images. To address this issue, the bilateral filtering method [34] is employed. Bilateral filtering, an extension of Gaussian filtering, incorporates weights related to pixel values. When the pixel values within the neighborhood differ significantly from the central pixel, the corresponding weight decreases. This approach effectively maintains the edge features of the facial image while eliminating high-frequency details. Specifically, for a filtering window centered at position q, consisting of a set of pixels S, the filtered result I_{BF} is computed as follows:

$$I_{BF} = \frac{1}{W_q}\sum_{p\in S} exp(-\frac{||p-q||^2}{2\sigma_s^2})exp(-\frac{||I_p-I_q||^2}{2\sigma_r^2})I_p \tag{10}$$

Where, σ_s and σ_r are user-specified parameters, I_p and I_q represent the pixel values corresponding to positions p and q, respectively. W_q represents the sum of weights within the filtering window:

$$W_q = \sum_{p\in S} exp(-\frac{||p-q||^2}{2\sigma_s^2})exp(-\frac{||I_p-I_q||^2}{2\sigma_r^2}) \tag{11}$$

To generate the color stroke image, the image is first processed using median filtering with a window size of 5. Then, bilateral filtering is applied iteratively for 20 rounds with σ_s and σ_r values set to 7 and 25, respectively.

Nevertheless, some regions exhibit smooth transitions between different colored areas after filtering, posing challenges for users aiming to replicate specific effects. To tackle this issue, we employ a pixel value clustering technique based on semantic information. Initially, the facial image undergoes semantic segmentation. Within the hair area, clustering allows all pixel colors to be replaced by the nearest cluster center color, thereby eliminating smooth transition regions. The same methodology is applied to the facial area, yielding the final color stroke image.

7. Additional Results

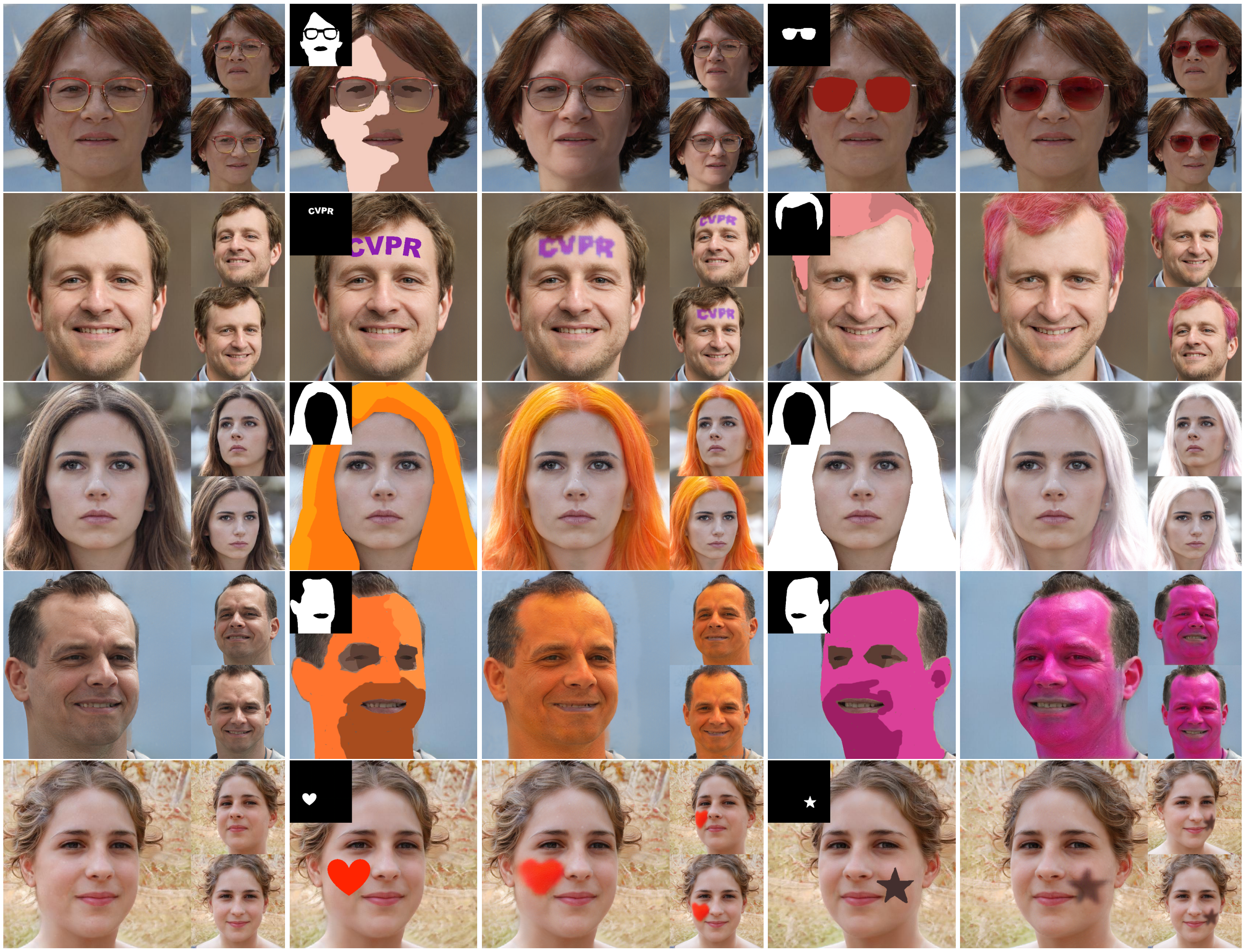
Synthetic Image Appearance Editing Results. Figure 9 presents additional appearance editing results produced by our methods. We randomly sampled latent codes from EG3D and performed diverse color stroke editing. Our model demonstrates its remarkable capability in flexibly modifying various appearance aspects from different angles, including lightning, hair styles, skin attributes, glasses, and facial patterns.

Real Image Appearance Editing Results. Figure 10 provides more results of real image appearance editing through Pivotal Tuning Inversion (PTI) [29] and our approach. This pipeline allows diverse and high quality editing over real facial image and generates corresponding 3D reconstruction. It holds significant promise for a variety of applications such as personalized user-driven 3D avatar editing.

Latent codes interpolation. Figure 11 provides linear interpolations between latent codes of distinct appearances produced by our method. The results validate that our 3D GAN inherits the well-behaved latent space of the StyleGAN2 [22] backbone and 3D-consistent editing results.

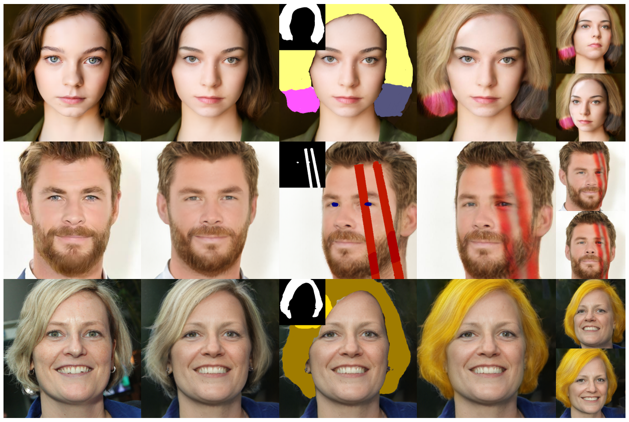
8. User Interface Design.

As shown in Figure 12, we design a user interface on top of the proposed pipeline to facilitate color stroke-based free view facial editing using [2]. Users can edit facial images randomly sampled from EG3D or upload real face images. Our system then synthesizes and displays photo-realistic 3D faces in the right panel. Users can change the views with the sliders on the control panel, and the 3D faces are rotated simultaneously. After editing from a single viewpoint,



(a) Original image (b) Input 1 (c) Results 1 (d) Input 2 (e) Results 2

Figure 9. Additional color stroke-based facial NeRF appearance editing results. Given the original faces (a), users drawn the editing color strokes (b) (d). Our method generates the corresponding editing results (c)(e). Additional masks are shown in top left corner, which is the same as stroke's shape in the first four rows, and drawn by users in the last row.



(a) Real image (b) Inversion (c) Stroke (d) Result

Figure 10. Additional real image appearance editing results.

users can also rotate the face, engaging in continuous editing from alternative views, thereby supporting detailed and expressive facial appearance manipulation. Throughout the editing process, users have the flexibility to erase or reset strokes and draw new strokes to depict their desired appearance. Our system defaults to inferthe mask area, denoted as M, based on users' edits over the stroke images. Additionally, for users who prefer a more localized editing, we have included a tab for drawing masks to specify particular areas of interest. This offers users greater control over the editing process, allowing for both global and local modifications to achieve the desired facial appearance.

9. Implementation Details

We implemented our framework by building upon the official PyTorch implementation of EG3D [6] available at https://github.com/NVlabs/eg3d and the official implementation of the pSp [28] network, accessible at https://github.com/eladrich/pixel2style2pixel. We adopt the

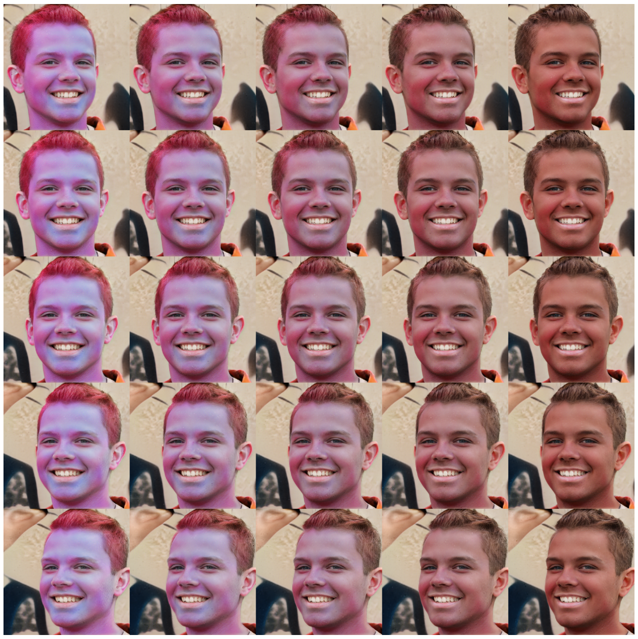
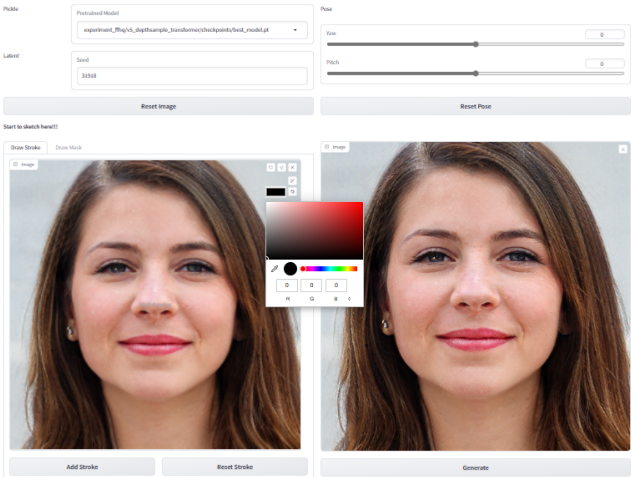


Figure 11. Latent codes interpolation. We linearly interpolate between latent codes representing distinct appearances of an individual. In each row, we keep the camera pose constant and interpolate the latent codes. In each column, we fix the latent code and show the rendering results of different camera angles.



Pickle

Pretrained Model: experiment_ffhq/v5_depthsample_transformer/checkpoints/best_model.pt

Latent

Seed: 31918

Pose

Yaw: 0

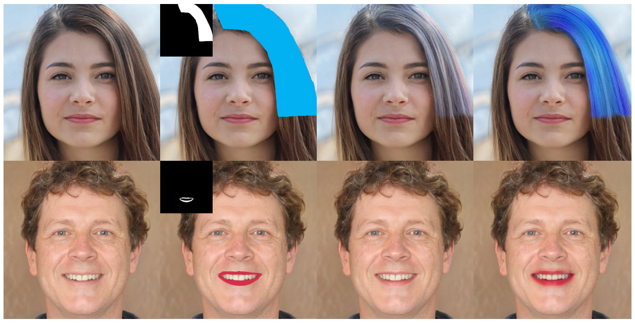
Pitch: 0

Start to sketch here!!!

Figure 12. The user interface of StrokeFaceNeRF. The interface empowers users to apply color strokes to generated images or upload their own images. The resulting edited images are displayed in the right panel. A control panel positioned at the top of the interface facilitates various key operations, encompassing the model weights, manipulation of random seed, and control over viewpoint rotation.

hierarchical encoder network architecture from pSp as our color encoder, and the EG3D generator is employed to generate the final facial NeRF based on the fused latent codes and mask inputs.

Latent Fusion Network. Our latent fusion network is a cross-attention transformer block, customized for improved fusion between the color stroke latent code w_c and the original facial latent feature w. Specifically, the input query tokens Q are derived as $w_c W_Q$, while the key tokens K and value tokens V are projected from w as wW_K and wW_V respectively. To mitigate the potential influence of the original latent code, we employ a routing scheme inspired by [14] to align the value features with queries uniquely. More precisely, the Softmax layer $Softmax_Q$ normalizes over queries instead of keys and subsequently re-normalizes the matrix over keys using $Norm_K$. The final edited latent code w_{edit} is calculated as follows:



(a) Original (b) Input (c) w/o 2stage (d) Ours

Figure 13. Additional ablation study results.

$$w_{edit} = Norm_K(Softmax_Q(\frac{QK^T}{\sqrt{d}}))V \qquad (12)$$

Mask Padding. Our volumetric mask fusion method effectively facilitates local editing and preserves original features in unedited regions through a 2D mask input. However, challenges arise during significant viewpoint changes, as depicted in Figure 14. To tackle this issue, we employ a mask padding technique. When projecting the 2D mask onto the 3D space using a user-specified camera pose, we introduce directional offsets to the camera positions, specifically setting them to 0.2 for upward, downward, leftward, rightward, and diagonal directions. For mask boundaries aligned with the original image's boundary, we extend the boundary pixels into newly offset regions, synthesising an expended mask image. These new augmented masks are then projected into 3D space using the offset camera poses, and the resulting point sets are merged to construct the mask volume. This adjustment enhances the completeness of the 3D point cloud obtained from the projection of the 2D mask, thereby improving 3D consistency.

10. Experiment Details

10.1. Baselines

SC-FEGAN [19] is based on an image completion approach, using color strokes as control conditions to fill the editing area. We utilized the official code at https://github.com/run-youngjoo/SC-FEGAN and its pretrained model weights.

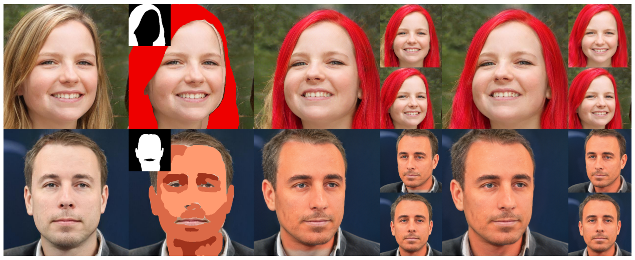


Figure 14. Mask padding editing results.

SDEdit [23] employs a diffusion model, guided by user-drawn color strokes to fill the editing areas based on a pre-trained generative model. We utilized the official code at https://github.com/ermongroup/SDEdit and its pretrained model weights.

pSp [28] is an efficient 2D image-to-image translation framework based on the StyleGAN generator. To establish a baseline for 3D stroke-based editing, we retrained a modified pSp network using the official codes on our datasets. This network directly projects the drawn color stroke images into EG3D's latent space, obtaining 3D facial results represented by latent codes. To better preserve the original facial features, another baseline, denoted as pSp-mixing, applies style-mixing techniques on pSp's predicted latent code by replacing layers 7-14 with the original latent code, then produces the final facial NeRF.

10.2. Additional Ablation Study

To justify our proposed 2 stage training strategy and the architecture of latent fusion network, we conduct additional ablation experiments. Both baseline approaches are trained on the same datasets with our approaches.

As depicted in Figure 13 (c), the removal of the two-stage training strategy, where both modules are trained simultaneously, poses challenges to network optimization, leading to incomplete effects in color stroke editing, and the correspondence with input strokes greatly deteriorates.

11. Discussion

Despite the success of our method in achieving high-quality color editing through color strokes, it is essential to address certain limitations that can impact its performance. Firstly, when it comes to highly intricate regions such as eyeballs, our method sometimes encounters challenges in producing optimal editing results. To overcome this limitation, implementing specialized processing for these detailed areas, possibly through segmentation, could significantly enhance the editing accuracy and effectiveness.

Additionally, the constraints imposed by our training dataset present difficulties in handling exceedingly complex color strokes, such as rainbow-colored hair. In such cases, our method tends to average these strokes, resulting in a potential loss of intricacies. A potential avenue for improvement involves the generation of distinct colors independently, coupled with subsequent mask creation and blending, to achieve more nuanced and refined editing results.

12. Ethical Discussion

While our work primarily focuses on positive applications such as virtual avatar design and user customized photo retouching, we acknowledge the importance of addressing ethical implications. We are committed to promoting positive usage of our technology and advocate for ethical considerations in its application. we do not condone the potential misuse of the facial editing tools. Many works [**? ? ?**] in the fake detection community could discriminate between the synthesized and real faces, and partly prevent the misuse.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4432–4441, 2019. 6

[2] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019. 11

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 4

[4] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020. 2

[5] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 1, 2, 3, 12

[7] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 6

[8] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[9] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*. Wiley New York, 1973. 6

[10] Lin Gao, Feng-Lin Liu, Shu-Yu Chen, Kaiwen Jiang, Chun-Peng Li, Yu-Kun Lai, and Hongbo Fu. Sketchfacenerf: Sketch-based facial generation and editing in neural radiance

fields. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2023)*, 42(4):159:1–159:17, 2023. 3

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[12] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022. 1

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[14] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11337–11346, 2022. 4, 13

[15] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1510–1519. IEEE Computer Society, 2017. 3

[16] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, and Feiyue Huang Jilin Li. Curricularface: Adaptive curriculum learning loss for deep face recognition. pages 1–8, 2020. 6

[17] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Lin Gao. Nerffaceediting: Disentangled face editing in neural radiance fields. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 3

[18] Kyungmin Jo, Gyumin Shim, Sanghun Jung, Soyoung Yang, and Jaegul Choo. Cg-nerf: Conditional generative neural radiance fields for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 724–733, 2023. 2, 3

[19] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1745–1753, 2019. 2, 7, 13

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 5

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 2, 3, 11

[23] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2, 7, 14

[24] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1

[25] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 3

[26] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 3

[27] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972*, 2018. 2

[28] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 2, 4, 5, 7, 12, 14

[29] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 4, 7, 11

[30] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2017. 2

[31] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1, 3

[32] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *ACM Transactions on Graphics (ToG)*, 41(6):1–10, 2022. 2, 3

[33] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 2, 3

[34] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998. 4, 11

[35] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 2

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[37] Chufeng Xiao, Deng Yu, Xiaoguang Han, Youyi Zheng, and Hongbo Fu. Sketchhairsalon: Deep sketch-based hair image synthesis. *arXiv preprint arXiv:2109.07874*, 2021. 2

[38] Hongwei Yong, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. Gradient centralization: A new optimization technique for deep neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 635–652. Springer, 2020. 6

[39] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129: 3051–3068, 2021. 2, 4