

FED-NeRF: Achieve High 3D Consistency and Temporal Coherence for Face Video Editing on Dynamic NeRF

Hao Zhang

HKUST

hzhangcc@connect.ust.hk

Yu-Wing Tai

Dartmouth College

yu-wing.tai@dartmouth.edu

Chi-Keung Tang

HKUST

cktang@cs.ust.hk

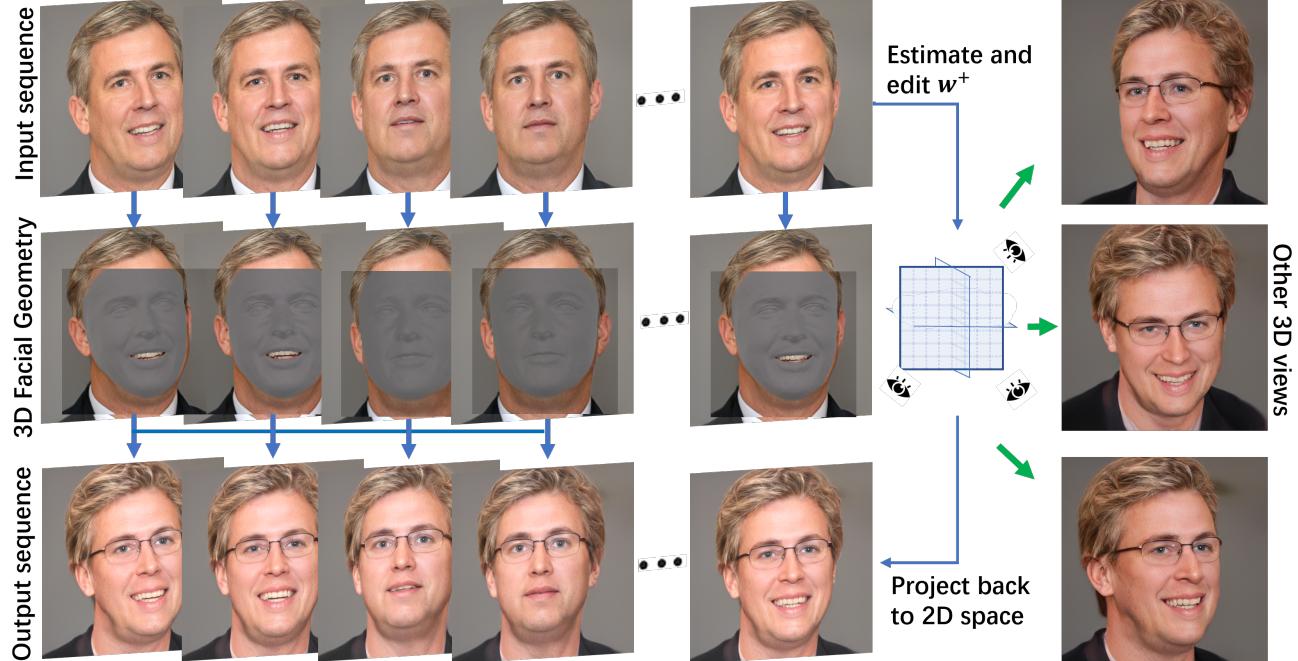


Figure 1. **Face video Editing results.** Editing prompts are “Wear a pair of glasses” and “Curly hair”. Every frame within the output sequence is rendered via the dynamic NeRF, which is precisely controlled by the estimated 3D facial geometry. Furthermore, the other 3D views effectively showcase the consistency of the dynamic NeRF.

Abstract

The success of the GAN-NeRF structure has enabled face editing on NeRF to maintain 3D view consistency. However, achieving simultaneously multi-view consistency and temporal coherence while editing video sequences remains a formidable challenge. This paper proposes a novel face video editing architecture built upon the dynamic face GAN-NeRF structure, which effectively utilizes video sequences to restore the latent code and 3D face geometry. By editing the latent code, multi-view consistent editing on the face can be ensured, as validated by multiview stereo reconstruction on the resulting edited images in our dynamic NeRF. As the estimation of face geometries occurs on a frame-

by-frame basis, this may introduce a jittering issue. We propose a stabilizer that maintains temporal coherence by preserving smooth changes of face expressions in consecutive frames. Quantitative and qualitative analyses reveal that our method, as the pioneering 4D face video editor, achieves state-of-the-art performance in comparison to existing 2D or 3D-based approaches independently addressing identity and motion. Codes will be released.

1. Introduction

Realistic human face synthesis and editing have constituted a prominent research area with their vast range of applica-

tions. Previous research employed subspace deformation or face morphing techniques [6, 41, 51] to achieve expression transfer and reenactment with impressive results. However, these methods were limited to blending existing faces and were unable to add substantially new or alter facial features realistically while preserving identity. With the debut of Generative Adversarial Networks (GANs) [10], the latent space of GANs, which possesses desirable properties such as perceptual path length and linear separability as elaborated in [13], have been employed to make face editing more flexible and versatile. Several recent studies, including [26, 37, 46], have demonstrated how to edit a face image in GAN’s latent space.

When editing a face in a *video*, possibly captured in the wild, it is crucial to ensure temporal coherence and 3D view consistency. To achieve temporal coherence, researchers in [1, 12, 43, 49, 52] have extended image editing to video editing by adding constraints between consecutive frames. Meanwhile, to ensure 3D view consistency while editing face features, researchers in [38, 39, 54] have leveraged the GAN-NeRF structure [4, 23, 33]. However, these methods are unable to simultaneously and robustly guarantee 3D view consistency and temporal coherence.

Thus it is quite imperative to elevate the editing process to a 4D space to achieve spatio-temporal coherence. Dynamic NeRF encompasses two desirable mechanisms in this regard: the first consists of a canonical space and a deformation space as outlined in various studies [24, 25, 27, 28], while the second involves conditioning the original neural radiance fields on time-related variables [44, 47, 56]. To attain better disentanglement of shape and motion, the first mechanism is adopted into our dynamic NeRF representation. In order to leverage the inherent editability within the latent space, the GAN-NeRF model incorporating the first mechanism [45, 48] emerges as a better option. Notably, the study by [48] uses the FLAME model [20] to represent the geometry of the deformation field, which offers higher expressiveness compared to 3DMM [2] used in [45].

The studies in [45, 48] proposed a structure for animating a talking head using a given latent code and a sequence of continuous expression codes. However, in video cases, obtaining the ground truth latent and expression codes remains challenging. Although GAN-inversion methods can generate the latent code given the estimated expression codes provided by off-the-shelf expression estimators [7, 9, 42], the resulting edited video may not achieve satisfactory performance due to inaccuracies accumulated during the estimation process.

To address editable dynamic face NeRF with the above issues, we introduce FED-NeRF, a novel face video editing architecture that thoroughly utilizes the information embedded in video sequences to restore the latent code of GAN-NeRF space, and the sequences of expression codes for each

frame as well. To accurately restore the latent code, we first extract w^+ features using an encoder based on [53] as the backbone for different frames. Next, we apply a cross-attention layer to these w^+ features to aggregate them into a single w^+ output. To predict sequences of expression codes, we modify the FLAME encoder of EMOCA [7] and incorporate it into the Omniavatar backbone [48] as the FLAME estimator during the training process. Since FLAME controls are estimated on a frame-by-frame basis, we introduce an algorithm that leverages the differentiability of the Catmull–Rom spline to stabilize the sequence of FLAME controls. Together with the edited w^+ by our modified Latent mapper, the edited video sequences can be produced. In summary, our main technical contributions are:

- We propose a latent code estimator that utilizes multi-frames as input and predicts accurate w^+ values that are applicable across a wide range of 3D views and face expressions.
- We propose a 3D face geometry estimator that accurately extracts face shape, expressions, and neck rotations from video sequences.
- We propose an algorithm that can effectively stabilize the transition of face geometry between consecutive frames.
- We modify the Latent mapper introduced by StyleClip [26] to enable its seamless integration with the Omniavatar backbone. [48].

Consequently, with our new technical contributions, casual users can easily edit facial features and expressions within a large range using simple prompts, while preserving the face’s identity and the rest of the given video, producing natural video results using the proposed editable 4D face NeRF. Moreover, as 3D consistency is naturally guaranteed, where the edited images can be immediately used in multi-view stereo for 3D face reconstruction. See Figure 1.

2. Related Work

Video Editing in 2D space Generative Adversarial Networks (GANs) [10] contribute to arguably the first breakthrough in contemporary 2D image generative methods, among which StyleGAN [13] and its variants [15, 17] stand out due to their expressive and well-disentangled latent spaces. Editing a single image via the latent space has been analyzed and shown to be successful in [26] and [46]. The straightforward approach [43] for editing a video is frame-by-frame processing in the same editing direction in the latent space. However, the same editing step cannot guarantee coherence among frames across all the given features, especially for high-frequency facial textures such as beard, mustache, hair, etc. To enhance the temporal coherence and avoid shape distortions between frames, in [12] the StyleGAN2 [15] latent vectors of human face video frames are disentangled to decouple the appearance, shape, expression, and motion from identity. In [50] learning a temporally

compensated latent code was proposed, which found incoherent noises lie in the high-frequency domain can be disentangled from the latent space. To remove the inconsistency after attribute manipulation, an in-between frame composition constraint was adopted. In addition to GAN models, diffusion models have also been employed for face editing in video sequences. In [18] the authors proposed a video editing framework based on diffusion autoencoders, which can effectively decompose identity and motion features from a given video. Nonetheless, the fundamental limitation of 2D video editing lies in its disregard of 3D geometry information during the editing process. This neglect results in shape distortion and feature alteration in side views, as depicted in Fig. 6. Furthermore, achieving multi-view consistency cannot be achieved, as illustrated in Fig. 7 and Tab. 1.

Video Editing in NeRF space The Neural Radiance Field (NeRF) [22], an implicit neural representation, has become the predominant approach in 3D generation. This method offers several advantages, including continuity, differentiability, compactness, and exceptional novel-view synthesis quality, distinguishing itself from conventional, explicit and discrete mesh and point cloud techniques.

GRAF [32] combines implicit neural rendering with GAN to create a generalizable NeRF. PiGAN [3] employs SiREN [36] to condition the implicit neural radiance field on the latent space. While 3D consistency is assured, volumetric rendering necessitates substantial computation. With limited computation, the image quality of these methods remains inferior to that of state-of-the-art 2D GANs. Consequently, numerous recent approaches adopt hybrid structures. StyleNeRF [11] applies volume rendering in the early feature maps in low resolution, followed by upsampling blocks to generate high-resolution images. In contrast to employing volume rendering in early layers, EG3D [5] performs the operation on a relatively high-resolution feature map using a hybrid representation for 3D features generated by StyleGAN2 [15] backbone, named tri-plane, which can incorporate more information than an explicit structure such as voxel. Given these advancements, 3D human face reconstruction achieves multi-view consistency and high-quality 3D generation.

Utilizing high-quality GAN-NeRF generation models, editing 3D facial structures has become increasingly feasible and promising. In [38, 39] an interactive approach was adopted for editing 3D faces, allowing users to draw directly on 2D images. In [54] a method was introduced on diffusion models for semantically editing facial NeRFs based on a target text prompt. Although editing faces on NeRF ensures multi-view 3D consistency, temporal or 4D coherence remains an issue.

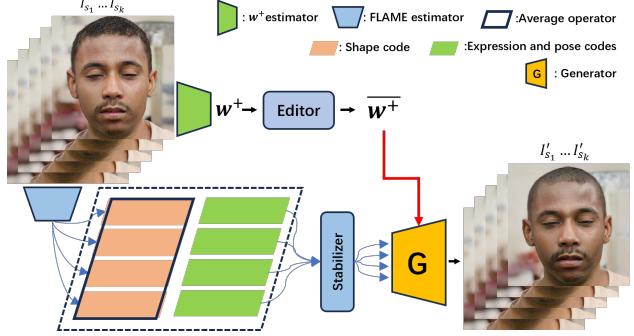


Figure 2. The Overview of our model. Given a video sequence, Our model will estimate a latent code w^+ and FLAME controls. The editor will subsequently modify the w^+ as \bar{w}^+ in accordance with a given text prompt. The Stabilizer then ensures the temporal consistency of the FLAME controls. Finally, the edited video sequence can be produced under the guidance of the stabilized FLAME controls and \bar{w}^+

3. Method

Figure 2 shows the overall framework. Given an input video, the Latent Code Estimator encodes the detailed face information embedded in the multiple frames into the latent code w^+ (Sec. 3.3). The Face Geometry Estimator extracts the shapes, expressions, and rotations of the jaw and neck from each frame (Sec 3.4). Since the face geometry is estimated individually on each frame, the Stabilizer is proposed to ensure coherence across frames (Sec. 3.5). In order to perform semantic editing of the facial features, we modify the Latent Code w^+ using the Semantic Editor (Sec. 3.6). Subsequently, with the integration of coherent facial geometries and a refined latent code, a photo-realistic edited video of exceptional fidelity can be produced. We use the Omniaavatar [48], a dynamic GAN-NeRF structure, as our Generator (Sec. 3.1). Our training and test data sets are described in Sec. 3.2.

3.1. Preliminaries

The Omniaavatar [48] utilizes a 3D-aware generator, EG3D [5], as the canonical space representation to achieve photo-realistic and multiview consistent image synthesis. Notably, Omniaavatar can disentangle control of face geometric attributes from image generation by employing a 3D statistical head model, FLAME [20]. Essentially, the pertinent deformation from the canonical space to the desired shapes and expressions is encapsulated by a trained deformable semantic SDF around the FLAME geometry. Specifically, a photo-realistic human face image $I_{RGB}(w^+ | c, p)$ can be generated by a given latent code w^+ , a camera pose c , and a FLAME control $p = (\alpha, \beta, \theta)$, which consists of shape code α , expression code β , jaw and neck pose θ .

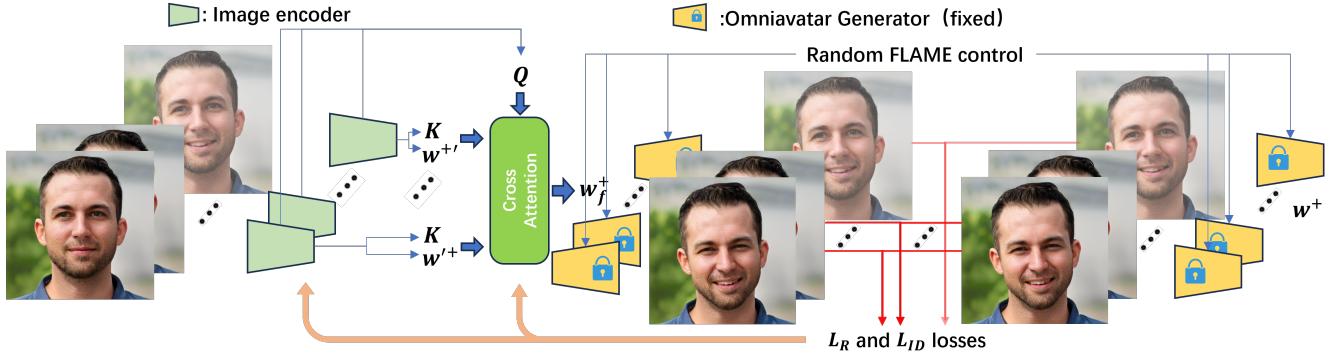


Figure 3. The structure of Latent Code Estimator. Given a video sequence, the Image encoder will extract features for each individual frame, which are then aggregated via the Cross Attention layer to produce a singular latent code output denoted as w_f^+ . The Losses \mathcal{L}_R and \mathcal{L}_{ID} are computed across multiple pairs of rendered images by utilizing the estimated w_f^+ and ground truth w^+ respectively.

3.2. Training data

Our objective is to reconstruct the w^+ and facial geometry from a video sequence. The majority of existing talking face datasets do not provide ground-truth facial geometry for each frame within the sequences. Thus, we utilize the Omniavatar [48] to synthesize multi-view images with various expressions for each subject. By randomly sampling n points from the Gaussian distribution and then transforming them through the mapping function, we obtain n latent codes $w_i^+, i \in [0, n - 1]$. For each w_i^+ , 60 FLAME controls P_i^0, \dots, P_i^{59} are randomly sampled from a large collection of 3D deformed FLAME datasets. Our training dataset \mathbf{D} is thus obtained, where we sample $n = 30,000$ as the training set, and $n = 300$ as the test set. As the FLAME control includes head rotation, we set all camera poses correspond to the frontal view:

$$\mathbf{D} = \{(w_0^+, I_{RGB}(w_0^+|c, p_0^0), \dots, I_{RGB}(w_0^+|c, p_0^{59})), \dots, (w_{n-1}^+, I_{RGB}(w_{n-1}^+|c, p_{n-1}^0), \dots, I_{RGB}(w_{n-1}^+|c, p_{n-1}^{59}))\}$$

3.3. Latent Code Estimator

To aggregate the identity information from the video sequence, we propose the Latent Code Estimator shown in Figure 3. Inspired by [53], for each frame, we extract the tuple (Q, K, w^+') according to the backbone’s pyramid features progressively. Here, we also choose Swin-transformer [21] as the backbone, and further add attention modules at different scale feature layers for different level latent codes, which are concatenated together as the w^+ output. The Q, K are extracted from the last layer of the pyramid features by MLP layers, since the Q, K contain the information of how to merge multiple w^+' s, which is high-level information and thus should be extracted from the latter layers of the pyramid features. After obtaining the tuples $(Q_i, K_i, w_i^+'), i \in [0, m]$, m is the number of the input frames, the $Q_i, i \in [0, m]$ are averaged to \bar{Q} , a Multi-

Head Cross Attention layer is applied on the tuples to get the final w_f^+

$$w_f^+ = \text{MultiHead}(\bar{Q}, \mathbf{K}, \mathbf{w}^+) \\ \text{where } \mathbf{K} = [k_0, \dots, k_m]^\top, \mathbf{w}^+ = [w_0^+, \dots, w_m^+]^\top$$

To fully disentangle the w^+ estimation with the facial geometry sample, t FLAME controls p_0, \dots, p_{t-1} are randomly sampled from the large collection of 3D deformed FLAME datasets, when calculating the following Loss function. As shown in Figure 3, the Reconstruction Loss \mathcal{L}_R and ID Loss \mathcal{L}_{ID} are used to measure the differences between the rendered images using w_f^+ and the rendered images using ground truth w^+ :

$$\mathcal{L}_R = \sum_i \left(\|V(I_{RGB}(w^+, p_i)) - V(I_{RGB}(w_f^+, p_i))\|_2^2 \right) \quad (1)$$

$$\mathcal{L}_{ID} = \sum_i \left(1 - \langle R(I_{RGB}(w^+, p_i)), R(I_{RGB}(w_f^+, p_i)) \rangle \right) \quad (2)$$

where $V(\cdot)$ is VGG16 image encoder [35], R is the pre-trained ArcFace network [8]. The total loss function is thus $\mathcal{L} = \mathcal{L}_R + \mathcal{L}_{ID}$. During training, the Omniavator Generator I_{RGB} is fixed and the image encoder and Cross Attention layer will be updated. Since the camera poses c remain fixed towards the frontal view, it is omitted from the above equations.

3.4. Face Geometry Estimator

Since current facial geometry predictors [7, 9] are not designed to deform the canonical space of a given dynamic NeRF, the results of directly applying these methods are distorted. To tackle this problem, we propose the framework depicted in Figure 4: the image encoder is modified based on [7], which first factorizes the input images into facial geometry (represented by FLAME controls), albedo, lighting, additional expression codes, etc. Given these factors, one

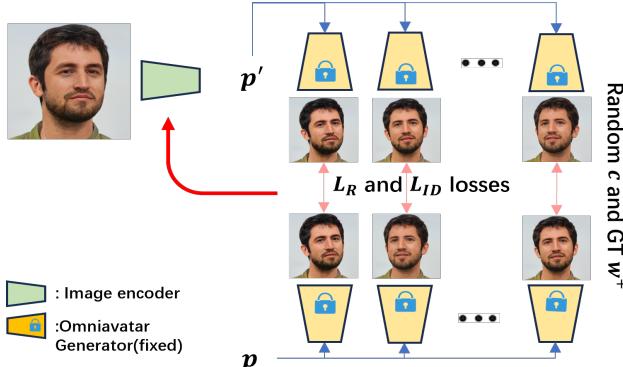


Figure 4. The structure of the Face Geometry Estimator. An estimation of the FLAME control p' can be obtained from an input image. Subsequently, pairs of images can be rendered with randomly sampled camera poses, and these losses can be computed based on these pairs.

can differentiably render an output image that should look similar to the input. As albedo, lighting, and detailed face texture are embedded in the latent code w^+ , only the attributes related to predicting facial geometry are kept while the rest of the EMOCA are discarded. After obtaining the FLAME controls, together with the latent code w^+ , an output image can be rendered by the Omniavatar Generator. To disentangle the facial geometry with the view direction of the input image, random camera poses c_i are used when rendering the output images. Here, w^+ , p' , and p denote the ground truth latent code, the estimated FLAME control, and the ground truth FLAME control.

$$\begin{aligned}\mathcal{L}_R &= \sum_i (\|V(I_{RGB}(w^+|c_i, p')) - V(I_{RGB}(w^+|c_i, p'))\|_2^2) \\ \mathcal{L}_{ID} &= \sum_i (1 - \langle R(I_{RGB}(w^+|c_i, p')), R(I_{RGB}(w^+|c_i, p')) \rangle)\end{aligned}$$

Similarly, the total loss function is $\mathcal{L} = \mathcal{L}_R + \mathcal{L}_{ID}$. During the training process, the Face Geometry Estimator will be updated, and the Omniavatar Generator I_{RGB} is fixed.

3.5. Stabilizer

Facial geometries are predicted individually on each frame in Sec. 3.4. Thus continuity among consecutive frames cannot be guaranteed. As human motions are largely continuous (second-order differentiable by Newton’s Law), we propose to use the Catmull–Rom spline to enforce smooth motion across the video sequence.

Denote $\mathcal{P} = [p_0, p_1, \dots, p_{n-1}]^\top$ as the FLAME controls estimated by Sec. 3.3 at the n timestamps in a video sequences. We aim to estimate a smooth motion sequence $\hat{\mathcal{P}} = [\hat{p}_0, \hat{p}_1, \dots, \hat{p}_{n-1}]^\top$ from \mathcal{P} . Note that $p_i, \hat{p}_i \in \mathbb{R}^{106}$ (100 dimensions for expression control and 6 dimensions for rotation control) and $\mathcal{P}, \hat{\mathcal{P}} \in \mathbb{R}^{n \times 106}$. We split the sequence \mathcal{P} into m sub-sequences $\mathcal{P}_0, \dots, \mathcal{P}_{m-1}$ with the length $\lfloor n/m \rfloor$ by selecting one FLAME control from ev-

ery m consecutive FLAME controls. $\mathcal{P}_0, \dots, \mathcal{P}_{m-1} \in \mathbb{R}^{\lfloor n/m \rfloor \times 106}$:

$$\begin{aligned}\mathcal{P}_0 &= [\hat{p}_0, \hat{p}_m, \hat{p}_{2m}, \dots, \hat{p}_{n-m}] \\ \mathcal{P}_1 &= [\hat{p}_1, \hat{p}_{m+1}, \hat{p}_{2m+1}, \dots, \hat{p}_{n-m+1}] \\ &\dots \\ \mathcal{P}_{m-1} &= [\hat{p}_{m-1}, \hat{p}_{2m-1}, \hat{p}_{3m-1}, \dots, \hat{p}_{n-1}]\end{aligned}\quad (3)$$

We use the Catmull–Rom spline to form m functions $f_0^i, f_1^i, \dots, f_{m-1}^i$ using the above m subsequences respectively for the dimension $i \in [0, 106 - 1]$. Thus, for dimension i , we obtain m estimations from the corresponding m functions at every timestamp. We calculate the distance $D_j^i(t)$ between an estimation $f_j^i(t)$, $j \in [0, m - 1]$ with the rest of the estimations at timestamp $t \in [0, n - 1]$.

$$D_j^i(t) = \sum_{j'=0, \dots, m-1} |f_j^i(t) - f_{j'}^i(t)| \quad (4)$$

The function f_j^i whose distance is ranked within the top two-thirds of $D_j^i(t)$, $j \in [0, m - 1]$ in ascending order will be averaged as the value of \hat{p}_t^i which denotes the value of the i th dimension of \hat{p}_t . The last one-third is regarded as outliers and will be discarded. After calculating \hat{p}_t^i for all of the dimensions and all of the timestamps, $\hat{\mathcal{P}}$ is computed. Algorithm 1 provides the detailed procedure. The hyperparameter m governs the degree of smoothness in the model. An increased value of m yields a more seamless transition, albeit at the expense of reduced fidelity, and vice versa. (Please note that the argmin operator will return a list of the indices in ascending order based on the value of $D_j^i(t)$.)

Algorithm 1 Stabilizer

Require: \mathcal{P} and a hyperparameter m

```

Initialize  $\hat{\mathcal{P}}$ 
Split  $\mathcal{P}$  into  $\mathcal{P}_0, \dots, \mathcal{P}_{m-1}$  according to (3).
for  $i = 0, \dots, 106 - 1$  do
    Form  $f_0^i, f_1^i, \dots, f_{m-1}^i$  by interpolating
     $\mathcal{P}_0^i, \dots, \mathcal{P}_{m-1}^i$  using Catmull–Rom spline.
    for  $t = 0, \dots, n - 1$  do
        for  $j = 0, \dots, m - 1$  do
            Compute  $D_j^i(t)$  according to (4).
        end for
         $l = \arg \min_{j \in [0, \dots, m-1]} D_j^i(t).$ 
         $l = l \left[ : \lfloor \frac{2m}{3} \rfloor \right]$ 
         $\hat{p}_t^i = \frac{1}{\lfloor \frac{2m}{3} \rfloor} \sum_{v=l[0], \dots, l[-1]} f_v^i(t)$ 
    end for
end for

```

3.6. Semantic Editor

We perform semantic editing in the w^+ space, where several off-the-shelf methods already exist [26, 34, 39, 46, 54].

In this study, we employ the Latent Mapper introduced in StyleClip [26], as it offers a short inference time of $75ms$ when pre-trained for a particular text prompt. The backbone of the StyleClip is the 2D StyleGAN [16]. In our work, the StyleGAN backbone is replaced by the Omniavatar Generator. Since all expressions are deformed with respect to the canonical space, which means once the canonical space is edited, all expressions of this person will be edited accordingly. Thanks to this property, our editing is 3D-view consistent and temporally coherent. Thus, the facial geometry control p can be set as zero which corresponds to the canonical space. The latent code w^+ is split into three groups (coarse, medium, and fine), or $w^+ = (w_c, w_m, w_f)$. We adopt the same structure of the Latent Mapper as StyleCLIP. \mathcal{L}_{CLIP} is modified as the following:

$$\hat{w}^+ = w^+ + M_t(w^+) \quad (5)$$

$$\mathcal{L}_{CLIP} = \mathcal{D}_{CLIP}(I_{RGB}(\hat{w}^+|c_0, p_0), t) \quad (6)$$

where the camera pose c_0 is towards the frontal face, p_0 is 0 , $M_t(\cdot)$ is the Latent Mapper, and $\mathcal{D}_{CLIP}(\cdot, \cdot)$ is the cosine distance between the CLIP embeddings of the input image and input text prompt [29]. The \mathcal{L}_{ID} is modified as the following:

$$\mathcal{L}_{ID} = 1 - \langle R(I_{RGB}(\hat{w}^+|c_0, p_0)), R(I_{RGB}(w^+, |c_0, p_0)) \rangle \quad (7)$$

The total Loss function is analogous to that utilized in StyleCLIP. For a comprehensive elucidation, please refer to the supplementary materials.

4. Experiments

4.1. Editing in-the-wild video sequences

Our method is capable of achieving good editing results for real-world cases, as demonstrated in Figure 5. This is due to the fact that Omniavatar [48] is trained on FFHQ [14], a human face dataset containing 70,000 in-the-wild images. The synthesized images generated by Omniavatar exhibit a distribution that is similar to that of real-world cases. Our approach provides greater flexibility in video editing compared to other methods [18, 43, 49, 52] due to the complete disentanglement between facial geometry and face semantic features. This allows us to explicitly edit facial expressions in a video by modifying the FLAME controls, whereas other methods are limited to semantic editing. Additional examples can be found in the supplementary materials.

4.2. Comparison

We conducted a comparative analysis between our proposed method and other existing techniques that aim to facilitate video editing. For ease of reference, we abbreviated the works [43], [49], and [18] as STIT, VideoEditGAN, and

DVA, respectively. To evaluate the performance of each method, we randomly selected a talking head video sequence from YouTube. As depicted in Fig. 6, the individual in the input sequence turns her face to the side, posing a challenge for all methods to generate a consistent editing outcome in response to the prompt “Wear a pair of glasses”. Notably, both STIT and VideoEditGAN, being 2D GAN-based methods that operate solely in the 2D GAN space and lack awareness of 3D information, fail to ensure the 3D consistency of the human face. As a result, they both produce frames in which the subject is not wearing glasses, or the eyeglasses deform unnaturally across different frames. In contrast, our proposed method exhibits superior performance in this scenario. To further investigate, we conduct Sec. 4.2.1 and Sec. 4.2.2 with respect to the 3D view consistency and temporal coherence. In addition to our superior performance, our method also boasts a significantly shorter inference time compared to others. This is due to our utilization of an encoder to locate the latent space, which allows for processing a video in just the time of one forward pass. In contrast, other methods require iterative algorithms such as GAN-inversion for STIT and VideoEditGAN, or diffusion iteration [30] for DVA. For a video sequence containing 100 frames, our processing time is approximately 3 minutes, whereas STIT and VideoEditGAN require around 3 hours, and DVA requires around 4 hours. All experiments were conducted using an RTX3090.

4.2.1 3D View Consistency

To examine the 3D consistency of edited views in an independent manner, a video capturing a static human face from continuously changing camera poses was selected for analysis. In Fig. 7, the left, frontal, and right views selected from such video are shown in the left column. After editing the video sequence respectively by VideoEditGAN, STIT, and our method, the 3D reconstruction error from the edited video sequence can be utilized as a metric for assessing the preservation of 3D consistency. To perform the multi-view stereo reconstruction, we employ COLMAP [31]. The quantitative outcomes are presented in Table 1. Our method yields the lowest mean reprojection error, a finding that is in line with the visualization results demonstrated in Figure 7. All the experiments are done on an RTX3090.

Table 1. Quantitative comparison on 3D consistency. Mean reprojection error of the COLMAP reconstruction.

| | STIT | VideoEditorGAN | Ours |
|----------------------|--------|----------------|---------------|
| Mean Reproj. Error ↓ | 1.1881 | 0.8488 | 0.7434 |



Figure 5. **More in-the-wild editing results.** These examples show that our model can achieve 3D consistency even when performing certain edits that alter the facial geometry, such as “Wear a pair of glasses”, “Short curly hair”, and so on.

4.2.2 Temporal Coherence

To quantitatively and qualitatively measure the temporal coherence, we select the video sequences from the CelebV-HQ dataset [55]. Raft [40] is used to estimate the optical flow between two consecutive frames, which serves as a metric for Please refer to the supplementary materials for visualization results and the statistics.

4.3. Reconstruction

As a sanity check, it is essential to reconstruct the original video from its encoded version. Failure to do so would result in losing the inherent identity of the original video before any editing can even be performed. Thus, we conducted this experiment on 5 randomly selected videos from the CelebV-HQ dataset [55]. The comparison is considered to be relatively fair, as none of the methods used in comparison have been trained on the CelebV-HQ dataset.

We choose the L2 distance between the feature encoded by VGG16 image encoder [35], denoted as \mathcal{L}_v . The quantitative comparison is shown in Tab. 2.

Table 2. **Quantitative comparison of reconstruction.** All the values in the table are multiplied by 100.

| | STIT | DVA | VideoEditorGAN | Ours |
|----------------------------|-----------|-----------|----------------|--------------------|
| $\mathcal{L}_v \downarrow$ | 0.47±0.21 | 0.36±0.07 | 0.68±0.45 | 0.31 ± 0.21 |

4.4. Ablation Study

We conducted an ablation study on the Latent Encoder Estimator. Specifically, we explored the following alternative setups: (a) employing a single frame as the input image during the training process, (b) selecting only one FLAME control to calculate the total loss \mathcal{L} during training, and (c) our full pipeline implementation, which uses five frames as the input images and five random FLAME controls to calculate the total loss \mathcal{L} . In this ablation, we randomly selected 100

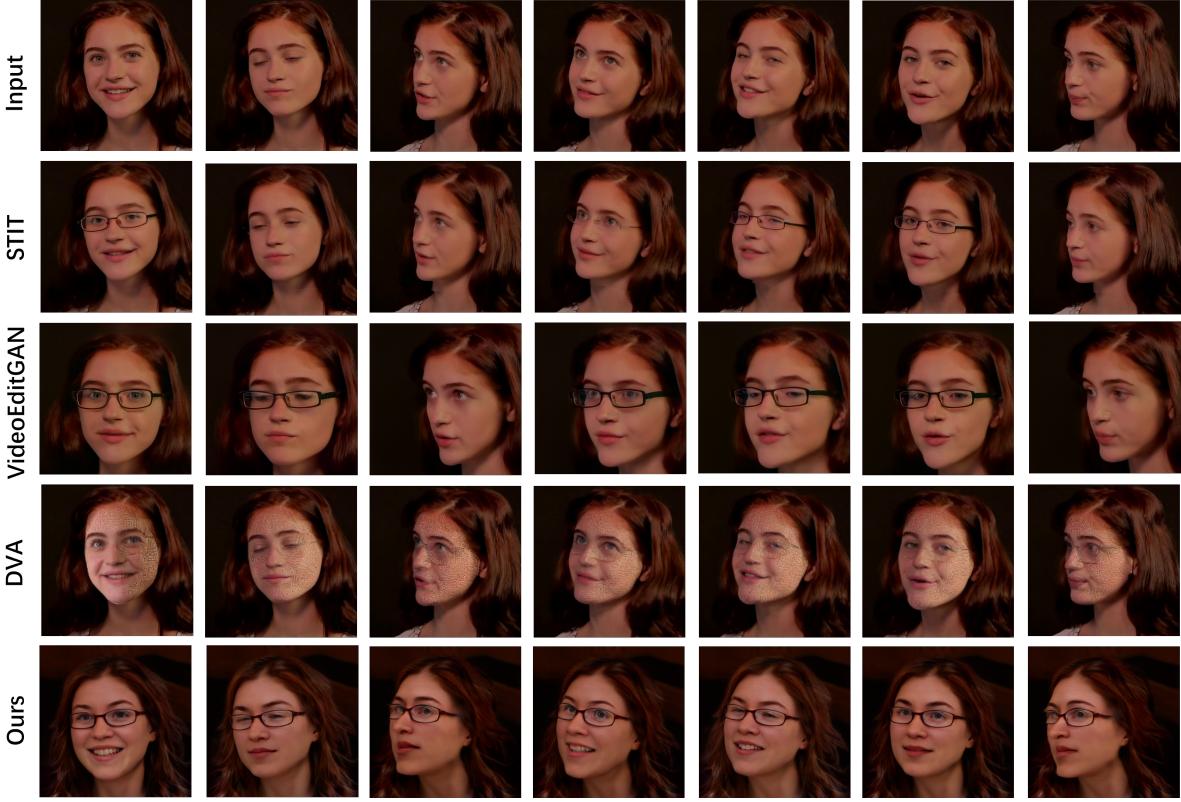


Figure 6. **Qualitative comparison.** The editing prompt is “wear a pair of glasses”. Note that our method achieves the highest level of 3D view consistency on the edited feature compared to other state-of-the-art methods. STIT, VideoEditGAN, and DVA are the abbreviation for [43], [49], and [18]

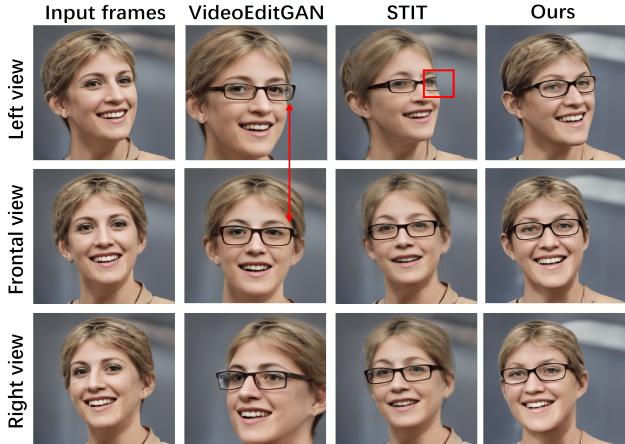


Figure 7. **Qualitative comparison of 3D consistency.** As indicated by the red box and arrow, it is evident that both STIT and VideoEditGan models are unable to generate 3D view consistency results of comparable quality to ours.

subjects from the test dataset. For each subject five random images were used as inputs for setups (b) and (c), while one of the five images is selected as the input for setup (a). Additionally, we selected another image distinct from the five images to calculate the \mathcal{L}_R and \mathcal{L}_{ID} . Table 3 tabulates the resulting statistics of this experiment. As indicated in the table, the full pipeline (c) outperforms other alternative setups

and achieves the best results.

Table 3. **Ablation study.** The mean value of \mathcal{L}_R and \mathcal{L}_{ID} are multiplied by 10. The standard deviation values of \mathcal{L}_R and \mathcal{L}_{ID} in the table are multiplied by 100.

| | (a) | (b) | (c) |
|-------------------------------|-----------------|-----------------|-----------------------------------|
| $\mathcal{L}_R \downarrow$ | 1.42 ± 3.41 | 1.40 ± 3.04 | 1.19 ± 3.10 |
| $\mathcal{L}_{ID} \downarrow$ | 0.28 ± 1.14 | 0.22 ± 0.92 | 0.16 ± 0.82 |

5. Conclusion and Discussion

Our proposed FED-NeRF elevates the face video editing process to operate in a 4D space, thereby ensuring both 3D view consistency and temporal coherence. To the best of our knowledge, this is the first work to tackle the video editing problem by utilizing Dynamic NeRF. Our novel latent Code Estimator utilizes a cross-attention mechanism to aggregate information embedded in multiple frames. The re-engineered Face Geometry Estimator and Stabilizer extract a sequence of facial geometries with good temporal coherence. Working together with our Semantic Editor, our approach presents a significant improvement compared to other video editors which may fall short in preserving geometry consistency across edited frames. We hope our work will inspire further research on solving the 2D video editing problem by incorporating the 4D world representation, which is more aligned with the spatio-temporal reality in

which we live.

References

- [1] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. *arXiv preprint https://arxiv.org/abs/2201.13433*, 2022. [2](#)
- [2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 187–194, 1999. [2](#)
- [3] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [3](#)
- [4] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proc. CVPR*, 2021. [2](#)
- [5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. *CoRR*, abs/2112.07945, 2021. [3](#)
- [6] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pages 1–10, 2011. [2](#)
- [7] Radek Dancek, Michael J. Black, and Timo Bolkart. EMOCA: Emotion driven monocular face capture and animation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20311–20322, 2022. [2, 4](#)
- [8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [4](#)
- [9] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 40(8), 2021. [2, 4](#)
- [10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. [2](#)
- [11] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenet: A style-based 3d-aware generator for high-resolution image synthesis. *CoRR*, abs/2110.08985, 2021. [3](#)
- [12] Yue-Ren Jiang, Shu-Yu Chen, Hongbo Fu, and Lin Gao. Identity-aware and shape-aware propagation of face editing in videos. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–12, 2023. [2](#)
- [13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. [2, 12](#)
- [14] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. [6](#)
- [15] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *CoRR*, abs/1912.04958, 2019. [2, 3](#)
- [16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [6](#)
- [17] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *CoRR*, abs/2106.12423, 2021. [2](#)
- [18] Gyeongman Kim, Hajin Shim, Hyunsu Kim, Yunjey Choi, Junho Kim, and Eunho Yang. Diffusion video autoencoders: Toward temporally consistent face video editing via disentangled video encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6091–6100, 2023. [3, 6, 8, 12, 13](#)
- [19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. [14](#)
- [20] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. [2, 3](#)
- [21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [4, 14](#)
- [22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. [3](#)
- [23] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#)
- [24] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. [2](#)
- [25] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), 2021. [2](#)
- [26] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, 2021. [2, 5, 6, 12](#)
- [27] Yicong Peng, Yichao Yan, Shengqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang

- Yang. Cagenerf: Cage-based neural radiance field for generalized 3d deformation and animation. In *Advances in Neural Information Processing Systems*, pages 31402–31415. Curran Associates, Inc., 2022. 2
- [28] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. *arXiv preprint arXiv:2011.13961*, 2020. 2
- [29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6
- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 6
- [31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6
- [32] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: generative radiance fields for 3d-aware image synthesis. *CoRR*, abs/2007.02442, 2020. 3
- [33] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2
- [34] Yujun Shen, Ceyuan Yang, Xiaou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans, 2020. 5
- [35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 4, 7
- [36] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *CoRR*, abs/2006.09661, 2020. 3
- [37] Kunpeng Song, Ligong Han, Bingchen Liu, Dimitris Metaxas, and Ahmed Elgammal. Diffusion guided domain adaptation of image generators. *arXiv preprint https://arxiv.org/abs/2212.04473*, 2022. 2
- [38] 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
- [39] 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, 5
- [40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 7, 13
- [41] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [42] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network, 2016. 2
- [43] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit H. Bermano, and Daniel Cohen-Or. Stitch it in time: Gan-based facial editing of real videos, 2022. 2, 6, 8, 12, 13
- [44] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis, 2021. 2
- [45] Yue Wu, Yu Deng, Jiaolong Yang, Fangyun Wei, Chen Qifeng, and Xin Tong. Anifacegan: Animatable 3d-aware face image generation for video avatars. In *Advances in Neural Information Processing Systems*, 2022. 2
- [46] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 2, 5
- [47] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video, 2021. 2
- [48] Hongyi Xu, Guoxian Song, Zihang Jiang, Jianfeng Zhang, Yichun Shi, Jing Liu, Wanchun Ma, Jiashi Feng, and Linjie Luo. Omniavatar: Geometry-guided controllable 3d head synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12814–12824, 2023. 2, 3, 4, 6
- [49] Yiran Xu, Badour AlBahar, and Jia-Bin Huang. Temporally consistent semantic video editing. *arXiv preprint arXiv: 2206.10590*, 2022. 2, 6, 8, 12, 13
- [50] Yangyang Xu, Shengfeng He, Kwan-Yee K. Wong, and Ping Luo. Rigid: Recurrent gan inversion and editing of real face videos, 2023. 2
- [51] Fei Yang, Eli Shechtman, Jue Wang, Lubomir D Bourdev, and Dimitris N Metaxas. Face morphing using 3d-aware appearance optimization. In *Graphics Interface*, pages 93–99. Citeseer, 2012. 2
- [52] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hel-lier. A latent transformer for disentangled face editing in images and videos. *2021 International Conference on Computer Vision*, 2021. 2, 6, 12, 13
- [53] Ziyang Yuan, Yiming Zhu, Yu Li, Hongyu Liu, and Chun Yuan. Make encoder great again in 3d gan inversion through geometry and occlusion-aware encoding. *arXiv preprint arXiv:2303.12326*, 2023. 2, 4, 14
- [54] Hao Zhang, Yanbo Xu, Tianyuan Dai, Yu-Wing Tai, and Chi-Keung Tang. Facednerf: Semantics-driven face reconstruction, prompt editing and relighting with diffusion models, 2023. 2, 3, 5
- [55] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. CelebV-HQ: A large-scale video facial attributes dataset. In *ECCV*, 2022. 7, 13

- [56] Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. Mofanerf: Morphable facial neural radiance field. In *European Conference on Computer Vision*, 2022. 2



Figure 8. More in-the-wild editing results. These examples show that our model can achieve 3D consistency even when performing certain edits that alter the facial geometry, such as “Wear a hat”, “Short curly hair”, and so on.

A. Discussion

To facilitate the process of face video editing in a 4D space, it is imperative to achieve complete disentanglement between facial geometry and face semantic features. This disentanglement enables seamless semantic editing of the face and explicit control over facial expressions simultaneously. In comparison to other video editing techniques such as [18, 43, 49, 52], our approach offers significantly more flexible control over the editing process, albeit at the cost of introducing some distortion to the identity. To the best of our knowledge, our method is the first to tackle the video editing problem using Dynamic NeRF, and it still can achieve comparable semantic editing results while also ensuring 3D consistency in realistic expression editing shown in Fig. 8, Fig. 9 and the Demo Video.

B. More Details on Semantic Editor

Analogous to the Latent Mapper architecture presented in StyleCLIP [26], distinct layers of the latent code w^+ contribute to varying degrees of detail in the generated image [13]. As a result, it is customary to categorize the layers into three distinct groups (coarse, medium, and fine): $w^+ = (w_c, w_m, w_f)$, and assign each group a specific portion of the (extended) latent vector. The corresponding mapper function can be expressed as follows:

$$M_t(w^+) = (M_t^c(w_c), M_t^m(w_m), M_t^f(w_f)). \quad (8)$$

The L_2 norm of $M_t(w^+)$ is used to maintain the visual characteristics of the input image. In conjunction with the loss functions $\mathcal{L}_{\text{CLIP}}$ and \mathcal{L}_{ID} introduced in the main paper, the total loss functions can be formulated as follows:



Figure 9. **The Demonstration of the ability to edit the facial expressions explicitly.** The leftmost displays the FLAME mesh, defined by the flame control with a revised value indicated on the left. The other parameters of the flame control maintain consistency with those estimated by our Face Geometry Estimator.

$$\mathcal{L}(w) = \mathcal{L}_{\text{CLIP}} + \lambda_{ID}\mathcal{L}_{ID} + \lambda_{L2}\|M_t(w^+)\| \quad (9)$$

where λ_{ID} and the λ_{L2} are the hyperparameters that regulate the strength of ID preservation and editability, respectively.

C. More Details on Temporal Coherence

In order to quantitatively assess the temporal coherence of the proposed method, the Raft algorithm [40] is employed to estimate a dense displacement field between two successive frames $\mathcal{I}_1, \mathcal{I}_2$. The dense displacement field denoted by (f^1, f^2) maps each pixel (u, v) in \mathcal{I}_1 to its corresponding coordinates $(u', v') = (u + f^1(u), v + f^2(v))$ in \mathcal{I}_2 . We compute the Euclidean displacement for each pixel $D(u, v) = \sqrt{(f^1(u))^2 + (f^2(v))^2}$. The mean Euclidean displacements among all pixels are denoted as $fl(1, 2)$, where the 1 and 2 indicate that the mean Euclidean displacement is computed on the first and second frames. To evaluate a video sequence, the initial 40 frames are utilized to compute the mean Euclidean displacements for the sequence, denoted as $flv(1, 40)$:

$$flv(1, 40) = \frac{1}{40-1} \sum_{i=1,2,\dots,39} fl(i, i+1) \quad (10)$$

We compared our method with STIT [43] and VideoEditGAN[49] by calculating the metric $flv(1, 40)$ on 5 randomly selected video sequences from the CelebV-HQ dataset [55]. The editing prompt is “Wear a pair of glasses”. The results are shown in Tab. 4

Table 4. **Quantitative comparison on Temporal Coherence.** STIT and VideoEditGAN are the abbreviations for [43] and [49].

| | STIT | VideoEditorGAN | Ours |
|-------------------------|--------|----------------|---------------|
| $flv(1, 40) \downarrow$ | 0.5687 | 0.3890 | 0.3249 |

D. Explicit Editing of Facial Expressions

Our approach provides greater flexibility in video editing compared to other methods [18, 43, 49, 52] due to the clean disentanglement between facial geometry and face semantic features. This allows us to explicitly edit facial expressions in a video by modifying the FLAME con-

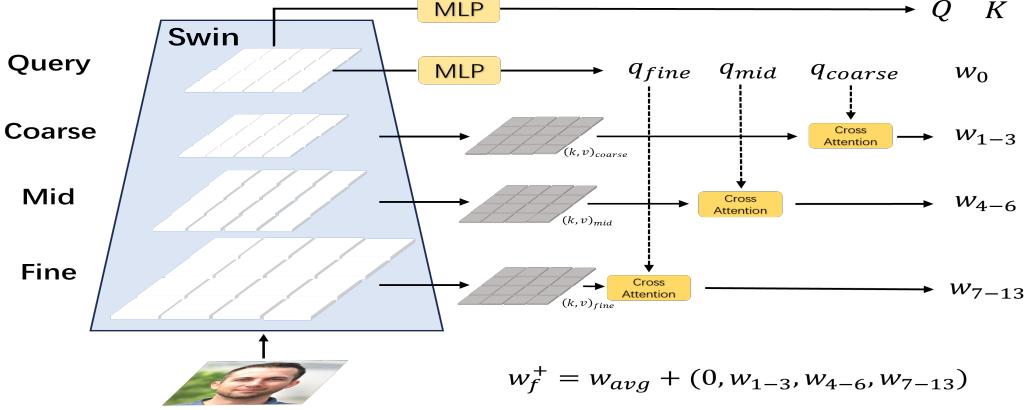


Figure 10. The structure of the Image Encoder of Face Geometry Estimator.

trols, whereas other methods are limited to semantic editing. Fig. 9 demonstrates that changing a value of the flame control can directly edit the facial expression.

E. Implementation Details

Latent Code Estimator To balance the overall performance and the GPU memory usage, 5 frames are randomly selected from training datasets and 5 different FLAME controls are used during calculating the loss. The structure of the image encoder is shown in Fig. 10. Inspired by [53], the intermediate output of the Swin-transformer [21] is split into four levels “query, coarse, mid, and fine”. “query” is used to get w_0 , q_{fine} , q_{mid} , and q_{coarse} . “coarse”, “mid”, “fine” layers are used to obtain keys and values $(k, v)_{\text{coarse}}$, $(k, v)_{\text{mid}}$, and $(k, v)_{\text{fine}}$. Then these level queries with their corresponding keys and values are sent into cross-attention layers to produce different w_i . The Q, K are extracted from the “query” layer by MLP layers, since the Q, K contains the information on how to merge multiple w^+ ’s, which is high-level information and thus should be extracted from the latter layers of the pyramid features. We trained the Latent Code Estimator on 2 V100 GPUs for nearly 2 weeks with batch size 2. The Adam optimizer [19] is used with an initial learning rate of 7×10^{-5} .

Face Geometry Estimator The learning rate is 5×10^{-5} which is decayed by 10^{-4} for each epoch. We trained the Face Geometry Estimator on 4 RTX3090s for 2 days with batch size 4.

Semantic Editor We trained the mapper with the following settings: $\lambda_{L2} = 0.7$, $\lambda_{ID} = 0.1$, maximum steps = 50000, batch size = 4, and learning rate $\in [1.0, 1.5, 2.0, 2.5, 3.0, 3.5]$. The learning rate depends on the editing test prompt. A large change in facial expressions usually needs a larger learning rate.

F. Demo Video

As the focus of our method is video editing, a demonstration video is a more effective means for evaluating the performance. Our demo video accompanying the supplementary material contains 1) comparisons with other methods, 2) explicit facial expression editing, and 3) in-the-wild video editing.