3D-Aware Face Editing via Warping-Guided Latent Direction Learning

Yuhao Cheng[1] Zhuo Chen[1] Xingyu Ren[1] Wenhan Zhu[1] Zhengqin Xu[1]
Di Xu[2] Changpeng Yang[2] Yichao Yan[1*]

[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Huawei Cloud Computing Technologies Co., Ltd

{chengyuhao,ningci5252,rxy_sjtu,zhuwenhan823,fate311,yanyichao}@sjtu.edu.cn,
{xudi21,yangchangpeng}@huawei.com

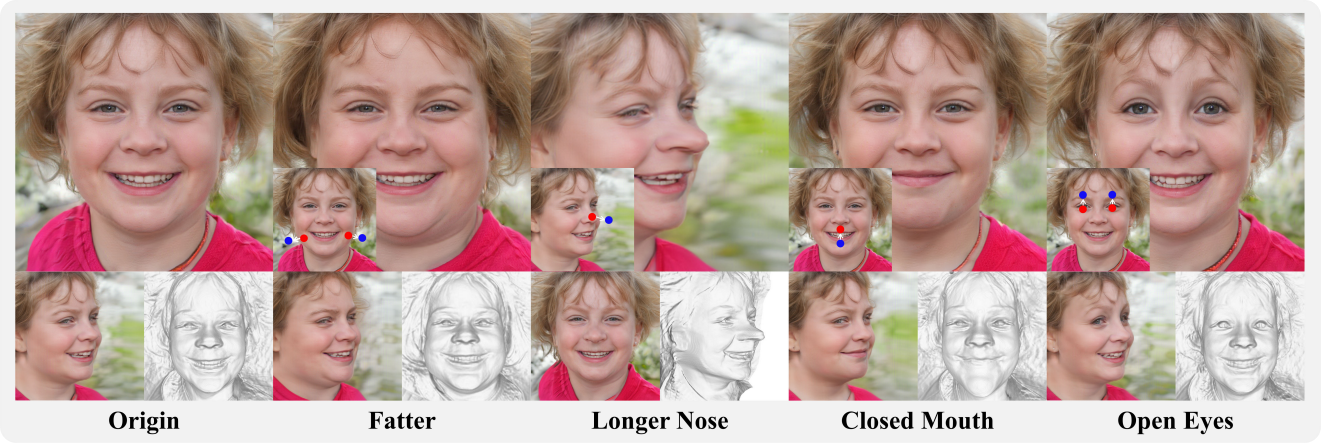


Figure 1. An example of our warping-guided 3D-aware face editing method. Our method supports users to edit 3D faces in an intuitive way that drags points from multiple perspectives. Moreover, our method can achieve disentangled editing for shape, expression, and view, while maintaining 3D consistency. Please **zoom-in** for detailed observation.

Abstract

3D facial editing, a longstanding task in computer vision with broad applications, is expected to fast and intuitively manipulate any face from arbitrary viewpoints following the user's will. Existing works have limitations in terms of intuitiveness, generalization, and efficiency. To overcome these challenges, we propose FaceEdit3D, which allows users to directly manipulate 3D points to edit a 3D face, achieving natural and rapid face editing. After one or several points are manipulated by users, we propose the tri-plane warping to directly deform the view-independent 3D representation. To address the problem of distortion caused by tri-plane warping, we train a warp-aware encoder to project the warped face onto a standardized latent space. In this space, we further propose directional latent editing to mitigate the identity bias caused by the encoder and realize the disentangled editing of various attributes. Extensive experiments show that our method achieves superior results with rich facial details and nice identity preservation. Our approach also supports general applications like multi-attribute continuous editing and cat/car editing. The project website is https://cyh-sj.github.io/FaceEdit3D/.

*Corresponding author

1. Introduction

High-quality face editing has long been an important research topic in computer vision with a wide range of applications, including social media and film production. Previous methods [16, 36, 43] based on 2D GANs [22, 23] have demonstrated the capability of editing facial images with high-fidelity. Recently, benefiting from the impressive achievements of 3D-aware generative models, especially in generative digital human [2–4, 11, 15, 32, 33, 41, 45, 51, 53, 55, 56, 64], the field of 3D facial editing has further attracted significant interest due to its promising capacity of manipulating a 3D representation.

Typically, 3D face editing methods can be generally classified into three categories: prior-guided conditioning, parameter-space fine-tuning, and latent-space optimization, as summarized in Tab. 1. Specifically, prior-guided conditioning methods [18, 46–48] employ an additional well-

Scheme	Methods	Intuitiveness	Generalization	Efficiency
Conditional control	[18, 46, 48]	✓	✗	✓
Fine-tuned models	[6, 13, 59]	✓	✓	✗
Supervised directions	[1, 36, 43]	✓	✗	✗
Unsupervised directions	[16, 42, 67]	✗	✓	✗
	[34] (2D)	△	✓	✗
	Ours	✓	✓	✓

Table 1. Summary of 3D-aware face editing methods. △ indicates its instructions are somewhat ambiguous semantically.

designed conditioning module to introduce the control information, *e.g.*, semantic maps [18, 46] and 3DMM [48, 49], into the 3D-aware models. Although flexible, these models typically require a large number of face images with their control labels for training. Parameter-space fine-tuning methods [6, 13, 59] optimize the pre-trained generators given the target input, achieving zero-shot editing with the help of the large language-image model, *e.g.*, CLIP [38] or Stable Diffusion [39]. However, it is required to maintain a particular generator for each specific editing target, severely constraining their generalization.

Due to the rich distributions learned in the pre-trained generator, discovering the meaningful directions in the latent space allows for a wide range of editing without the need to modify the generator and dependence on a large amount of training data. According to the exploration of editing direction, latent-space optimization can be achieved in supervised and unsupervised ways. Supervised methods [1, 36, 43, 44] search the meaningful directions in the latent space by learning labeled data for each specific editing. However, these methods cannot be generalized beyond the training domain. In contrast, unsupervised methods [16, 42, 50, 65–67] discover out-of-domain directions by analyzing the distribution of the latent space. However, the editing directions in the latent space are typically not semantically intuitive for the users. Accordingly, introducing interactive guidance to bridge the gap between the latent space and the user's intuition becomes the main purpose of the unsupervised methods.

To achieve this, several works [12, 34] utilize manipulating points on 2D images to optimize latent code in an unsupervised way, achieving image editing intuitively. The most prominent method DragGAN [34] proposes motion supervision and point tracking to optimize the latent code in a self-supervised manner, showcasing its flexible and intuitive editing capabilities. Considering their success on 2D images, it would be highly desirable if we could also manipulate 3D points to edit a 3D facial representation. However, it is non-trivial to directly extend point dragging to 3D-aware facial editing, due to the following challenges. 1) These methods ignore the global 3D facial structure and only focus on the movements of specific points, potentially leading to exaggerated distortions. 2) These methods employ an inefficient approach to optimize the latent codes for image editing. Therefore, extending this procedure to 3D-aware generators fails to meet the demands of 3D interactive applications. 3) The controllability of point dragging is less precise and may cause ambiguous targets, *e.g.*, enlarging the shape of the mouth may lead the mouth to open.

To overcome these challenges, we propose FaceEdit3D to learn editing directions guided by 3D-consistent face-warping, realizing intuitive and rapid 3D-aware facial editing. **(1)** First, we propose tri-plane warping on the 3D representation to achieve accurate 3D-consistent facial editing, which allows us to sidestep inaccurate motion supervision. Further, we introduce 3D landmarks rather than arbitrary points as face prior to constrain the change in the normal face distribution. Although tri-plane warping allows for precise editing, it introduces slight facial distortions. **(2)** Hence, we train a warp-aware encoder instead of latent optimization to straightforwardly project the warped renderings into the standardized space, enabling fast and photo-realistic editing. Due to the complex semantic information in the latent space of 3D-aware generators, the obtained encoder suffers from inherent bias, resulting in a loss of details and identity shifting. **(3)** Therefore, we propose to learn the hierarchical directional editing in latent space, enabling disentangled face editing with identity and details preservation.

With all the designs above, we successfully introduce dragging-based edits into 3D face representations. Our work achieves an efficient and straightforward editing process which also enables the decoupling of facial expressions and shapes. Compared to other face editing approaches, our method offers a more intuitive bridge but avoids dependence on the 3D annotations. Extensive experiments have demonstrated the superiority of our method in intuitiveness, generalization, and efficiency for the task of facial editing.

The main contributions are summarized as follows:

- We design an efficient and straightforward 3D-aware face editing pipeline that is in line with the user's intuition.
- We propose to warp the face in the tri-plane feature level, enabling 3D-consistent face manipulation.
- We propose a warp-aware encoder to better identify the subtle changes and efficiently solve the problem of distorted face caused by the tri-plane warp.
- We propose directional editing in latent space, achieving disentangled facial editing with the preservation of identity and details.

2. Related Works

2.1. 3D-aware GANs

Inspired by the superiority of implicit representation [31], several attempts [2–4, 11, 15, 32, 33, 41, 45, 53, 55, 64] deploy radiance fields into generative models and thus en-

able 3D consistent image synthesis. The capability of learning 3D representations from unposed single-view 2D images only empowers these 3D-aware GAN models to gain wide interests and applications. However, partial 3D-aware GANs [3, 15, 32, 33, 41, 64] adopt full implicit representation that lacks pre-computed 3D features before the point sampling. As a consequence, they need to regenerate the 3D feature when given novel viewpoints, limiting the efficiency of them in interactive applications. To address this challenge, several works [2, 4, 45, 53] adopt hybrid representations that first generate view-independent features, and enable sampling points on these pre-computed features for novel view synthesis. Consequently, these methods can realize rapid generation and maintain the inherent 3D-consistent representation. Specifically, EG3D [4] introduces the light tri-plane representation into the generator to raise efficiency and further enhance the image quality. Considering its efficient representation and mature downstream techniques, we adopt the EG3D [4] as the base 3D-aware model to demonstrate the effectiveness of our methods.

2.2. Implicit Representations Deformation

The deformation of 3D implicit representation has long attracted wide focus, as it serves as the foundation of broad animation applications. Prior researches predominantly introduce an additional deformation field based on the original representation to modify the 3D points. Specifically, deformation fields can be implemented through proxy-based editing [14, 21, 35, 57], cage-based editing [17, 37, 54], and parametric prior-based editing [40, 52, 63], etc. Proxy-based editing learns a lightweight neural network to compute the translation and rotation of 3D points, enabling the deformation of original 3D coordinates. The cage-based methods establish a surrounding cage to fully cover up the original surface of an implicit representation and then modify the cage to deform the inherent surface. Parametric prior-based methods leverage the parametric models such as SMPL [29] and FLAME [27] as a prior condition of the deformation network to drive the implicit representations. However, all of these approaches need to optimize a controllable module for each specific object, lack of efficiency and generality. In contrast, our work provides a landmark-based way to directly edit the 3D representation without optimization and further compresses the 3D deformation into 2D feature planes to improve efficiency.

2.3. Face Editing in GANs

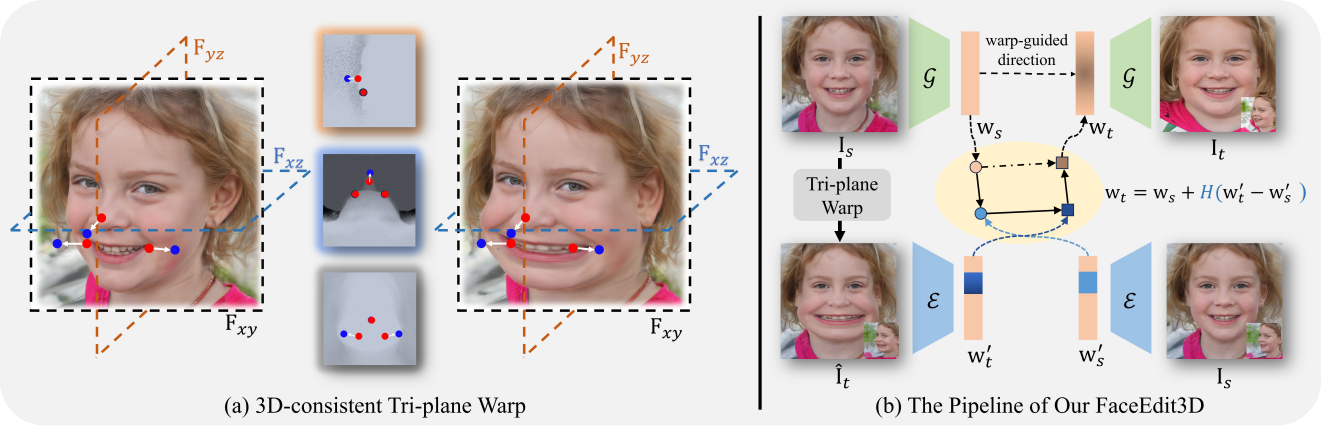
As the latent space learned by the conditioned GANs contains most of the distribution knowledge, many works [1, 42, 43, 50, 69] explore the latent space of a pre-trained generator for the following facial attribute editing. Specifically, InterFaceGAN [43] studies the semantics encoded in the latent space and disentangles the facial semantics with linear projection. To explicitly edit the facial attributes, further works explore utilizing the intuitive representation, *e.g.*, semantic maps [5, 46, 47, 68] and text prompts [19, 36] for the optimization or the extension of latent space. Moreover, an idea that directly drags the face for the editing catches the wide attention. DragGAN [34] optimizes the latent space via dragging selected points on the image to the target positions. However, it is hard to preserve the facial identity when setting a far distance between the two points, preventing the DragGAN from large-scale editing. Despite the prominent performance of latent space manipulation, it still faces a challenge in balancing the identity preservation and editing amplitude. To further enhance the editing capability, several works [6, 13, 24] focus on the parameter space of a pre-trained generator. While these methods can achieve out-of-domain editing, they need to maintain a specific generator for each attribute manipulation, lacking efficiency. Compared to the methods mentioned above, our method is an intuitive way of dragging points to deform the 3D representations while improving the efficiency and preserving the identity.

3. Methods

Our proposed framework, **FaceEdit3D**, aims at multi-view consistent facial editing in shape, expression, and pose via warping-guided directional editing, as illustrated in Fig. 2. To this end, we first review the 3D-aware GAN that achieves high-resolution face rendering from multiple views (Sec. 3.1). Based on the 3D-aware generator, we propose a point-guided feature-space warping method that manipulates the inherent tri-plane representations while ensuring the 3D consistency (Sec. 3.2). However, directly editing the tri-plane may lead to distortions in the final rendered images. Therefore, we train a specifically designed encoder to project the warped renderings to the standardized latent space for photo-realistic editing results (Sec. 3.3). Finally, we delve into the mechanism of latent space and propose directional editing in latent space that enables the disentangled editing of facial shape, expression, and pose (Sec. 3.4).

3.1. Preliminaries on 3D-aware Face Generator

Our framework is built upon EG3D [4], one of the most powerful 3D-aware generative models that achieve photo-realistic 3D face generation. The generator of EG3D introduces a tri-plane representation, which compactly encodes the geometry and appearance of a 3D face. Specifically, the tri-plane features can be denoted as $\mathbf{F} = \mathcal{G}(\mathbf{w}) \in \mathbb{R}^{3\times32\times256\times256}$, where $\mathbf{w}$ is a latent code. To render face images from a specific viewpoint, the features of 3D coordinates are sampled from the tri-plane features and a shallow decoder is leveraged to project the tri-plane feature $\mathbf{F}(x, y, z) \in \mathbb{R}^{32\times3}$ into volume density $\sigma \in \mathbb{R}^1$ and color feature $c \in \mathbb{R}^{32}$. Subsequently, a low-resolution fea-



(a) 3D-consistent Tri-plane Warp (b) The Pipeline of Our FaceEdit3D

Figure 2. Overview of our proposed FaceEdit3D. **(a)** A detailed illustration of our tri-plane warp. We project 2D key points onto the 3D face surface and then map them to each corresponding plane within a tri-plane representation. Afterward, we apply warping operations to each plane to achieve 3D-consistent editing. **(b)** The full pipeline of our FaceEdit3D. Given a source image $\mathbf{I}_s$ with its latent code $\mathbf{w}_s$, we first perform the tri-plane warping on it and obtain the warped rendering $\hat{\mathbf{I}}_t$. Subsequently, we utilize a warp-aware encoder to extract the latent codes $\mathbf{w}'_s$ and $\mathbf{w}'_t$ from the source image $\mathbf{I}_s$ and the warped renderings $\hat{\mathbf{I}}_t$, respectively. Then, we employ the hierarchical latent direction to update the target latent code $\mathbf{w}_t$. Finally, the edited facial image $\mathbf{I}_t$ can be synthesized via the updated latent code $\mathbf{w}_t$.

ture map is generated via volume rendering and then upsampled to high-resolution images. The representation ability of tri-plane features has been verified by several recent works [7, 20, 24]. Therefore, to achieve 3D-consistent editing, we choose to operate directly on the tri-plane features.

3.2. Multi-view Consistent Face Warping

For 3D face editing, it is a flexible way for users to directly drag points on the rendered images. Different from 2D-level editing that limits to one specific viewpoint, 3D-level manipulation should support editing from an arbitrary viewpoint and achieve 3D-consistent editing effects. To achieve this, we propose a framework based on point-guided tri-plane warping, where users manipulate one or several points from a desirable viewpoint, and the tri-plane features are warped according to the point displacements.

Point Manipulation by Users. Ideally, users can directly modify arbitrary points in a rendered face to achieve editing. Nevertheless, the potential conflicts among excessive control points may lead to undesirable distortions of the facial structure during the joint point manipulation, consequently yielding results that deviate from realistic human appearances. To address this issue, we constrain the users to manipulate a set of meaningful 3D facial landmarks to guarantee a natural face structure.

Specifically, given a latent code $\mathbf{w}_s$ and a pre-trained EG3D generator $\mathcal{G}$, the portrait is first rendered in the front view with camera intrinsic $\mathbf{K}$. Then, 2D facial landmarks are detected by a pre-trained detector and projected on the facial surface to obtain 3D landmarks $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_n\} \in \mathbb{R}^{n\times 3}$, and $\mathbf{p}_i = \{\mathbf{p}_i^x, \mathbf{p}_i^y, \mathbf{p}_i^z\} \in \mathbb{R}^3$. Consequently, users can render images from an arbitrary viewpoint with extrinsic $\mathbf{R} \in \mathbb{SO}(3)$ and select any specific points for editing. Take the selected point $\mathbf{p}_i$ as an example, we set the movement of the point $\Delta\mathbf{p}_i$ is perpendicular to the rendering direction. The updated 3D point $\mathbf{p}'_i$ is represented as:

$$\mathbf{p}'_i = \mathbf{p}_i + \mathbf{R}^{-1}\mathbf{K}^{-1}\mathbf{Z}\Delta\mathbf{p}_i, \tag{1}$$

where $\mathbf{Z}$ is the depth of the selected point in the pose $\mathbf{R}$. After manipulating specific points within the facial structure, we obtain a set of new 3D landmarks $\mathbf{P}' = \{\mathbf{p}'_0, \mathbf{p}'_1, \cdots, \mathbf{p}'_n\}$.

Tri-plane Warping. After the users have manipulated the key points, we apply 3D warping on the tri-planes to edit the 3D representation. Individually considering each of the tri-plane features [7], we can extend the editing in 3D space onto three 2D planes to enhance efficiency. Therefore, we begin by projecting the 3D landmarks onto the three feature planes, and then individually apply a similar warping transformation on each of these feature planes, as illustrated in Fig. 2 (a). Take the xy-plane $\mathbf{F}_{xy}$ as an example, given n source projected points $\mathbf{P}^{xy} = \{\mathbf{p}_0^{xy}, \mathbf{p}_1^{xy}, \cdots, \mathbf{p}_n^{xy}\} \in \mathbb{R}^{n\times 2}$, $\mathbf{p}_i^{xy} = \{\mathbf{p}_i^x, \mathbf{p}_i^y\}$ and their target points $\hat{\mathbf{P}}^{xy} = \{\hat{\mathbf{p}}_0^{xy}, \hat{\mathbf{p}}_1^{xy}, \cdots, \hat{\mathbf{p}}_n^{xy}\}$, we employ thin-plate spline interpolation [9] to compute the grid sampler with:

$$g(\mathbf{q}) = \sum_{i=1}^{n} w_i \phi\left(\|\mathbf{q} - \hat{\mathbf{p}}_i\|\right) + \mathbf{v}^T\mathbf{q} + \mathbf{b}, \tag{2}$$

where $\phi(r) = r^2\log(r)$ is the kernel function and $g(\mathbf{q})$ provides the inverse mapping of the location $\mathbf{p}$ to the original plane coordinates $\mathbf{q}$. The parameters $\mathbf{v}, \mathbf{b}$ are the parameters to minimize a certain definition of curvature. Similarly, by applying such inverse mapping to all three planes, we complete the tri-plane warping and achieve the inherently

3D-consistent modification. Compared to the manipulation of the sampled 3D coordinate space [60, 62], our method directly manipulates the 3D representation, empowering to simultaneously edit from multiple viewpoints without additional steps.

3.3. Warp-Aware Encoding

After tri-plane warping, the editing results exhibit 3D consistent modification. However, directly applying warping operation on tri-plane features may not conform to the facial distribution in the latent space, leading to a severely distorted appearance. To solve this problem, our solution is to encode the distorted facial image $\hat{\mathbf{I}}_t$ into a standardized latent space that learns the natural counterpart $\mathbf{w}'_t$ of the distorted face with an encoder $\mathcal{E}$:

$$\mathbf{w}'_t = \mathcal{E}(\hat{\mathbf{I}}_t). \tag{3}$$

To train the encoder, we sample images from the pre-trained generator to generate image and latent code pairs. Specifically, the portrait $\mathbf{I}_s$ is generated from the randomly sampled latent code and the camera poses $\mathbf{c}$. Subsequently, the portrait $\mathbf{I}_s$ is projected to latent code $\mathbf{w}'_s$ by the encoder $\mathcal{E}$, and then the corresponding image $\mathbf{I}'_s$ is generated by the same frozen generator $\mathcal{G}$ and pose $\mathbf{c}$. The optimization objective of the encoder is the combination of L1 Loss, LPIPS loss [61], and identity loss [10]:

$$\mathcal{L}_o = \mathcal{L}_1(\mathbf{I}_s, \mathbf{I}'_s) + \mathcal{L}_{\mathbf{LPIPS}}(\mathbf{I}_s, \mathbf{I}'_s) + \mathcal{L}_{\mathbf{ID}}(\mathbf{I}_s, \mathbf{I}'_s). \tag{4}$$

Unfortunately, we find that the encoder trained with the aforementioned method poses difficulties in identifying subtle modifications due to the inherent complexity of 3D-aware generators. Hence, we further introduce the tri-plane warping as the data augmentation to enhance the overall perception of subtle edits. Similar to the above training pipeline, we apply the encoder onto the warped rendering $\hat{\mathbf{I}}_t$ to obtain the latent code $\mathbf{w}'_t$, thus generating its inverted image $\mathbf{I}'_t$. The loss is calculated between $\mathbf{I}'_t$ and $\hat{\mathbf{I}}_t$:

$$\mathcal{L}_w = \mathcal{L}_1(\hat{\mathbf{I}}_t, \mathbf{I}'_t) + \mathcal{L}_{\mathbf{LPIPS}}(\hat{\mathbf{I}}_t, \mathbf{I}'_t) + \mathcal{L}_{\mathbf{ID}}(\hat{\mathbf{I}}_t, \mathbf{I}'_t). \tag{5}$$

Besides, following GOAE [58], we utilize a discriminator $\mathcal{D}$ to ensure the latent codes $\mathbf{w}'_t$ and $\mathbf{w}'_s$ in the standardized latent space:

$$\begin{aligned} \mathcal{L}_d = &\mathbb{E}[f(\mathcal{D}(\mathbf{w}'_t)) + f(\mathcal{D}(\mathbf{w}'_s))] \\ &+ \mathbb{E}[f\left(-\mathcal{D}\left(\mathbf{w}_c\right)\right)] + \gamma||\nabla\mathcal{D}\left(\mathbf{w}_c\right)||^2, \end{aligned} \tag{6}$$

where $f(x) = -\log(1+\exp(-x))$, and γ is a hyper-parameter in R1 regularization. $\mathbf{w_c}$ are pre-sampled standardized latent codes by the frozen generator. The final objective linearly combines the aforementioned losses:

$$\mathcal{L} = \mathcal{L}_o + \mathcal{L}_w + \mathcal{L}_d. \tag{7}$$

After the training process, the edited rendering is projected into latent space and then passed to the generator to yield a more reasonable editing result in the target view $\mathbf{c}_t$:

$$\mathbf{I}_t = \mathcal{G}(\mathbf{w}'_t, \mathbf{c}_t). \tag{8}$$

3.4. Directional Editing in Latent Space

Warp-aware encoder solves the problem of severely distorted appearance caused by the tri-plane warp, however, it additionally introduces identity bias into the latent codes as the encoder cannot faithfully inverse faces. Besides, it is still hard to handle the ambiguity during the point-manipulation. Therefore, we here propose directional editing learning to further overcome these two challenges.

To begin with, we adopt the difference between the latent codes that are extracted from the images before and after warping by the encoder as the direction guidance. In this way, we mitigate the identity bias and bypass the problem caused by the encoder. Furthermore, we follow StyleCLIP [36] to explore the semantics of layers in the $W+$ latent space of EG3D [4], empowering our method with the disentangled editing of the expression and shape. According to the hierarchical mechanism, we can obtain free editing results by applying editing directions in the variant layers to the same warping facial image, successfully avoiding the ambiguity caused by the tri-plane warp.

The full pipeline is shown in Fig. 2 (b). Given a latent code $\mathbf{w}_s$ and the frozen EG3D generator $\mathcal{G}$, the facial tri-plane can be generated. Specifically, the warp-aware encoder projects these two images to standardized latent codes $\mathbf{w}'_s$ and $\mathbf{w}'_t$ with Eq. (3), respectively. The target edited latent code $\mathbf{w}_t$ can be calculated with:

$$\mathbf{w}_t = \mathbf{w}_s + H(\mathbf{w}'_t - \mathbf{w}'_s), \tag{9}$$

where $H(\cdot)$ is a feature selection module for disentangling latent direction. Finally, the modified portrait $\mathbf{I}_t$ can be rendered from any perspective $\mathbf{c}_t$ with $\mathbf{I}_t = \mathcal{G}(\mathbf{w}_t, \mathbf{c}_t)$.

4. Experiments

In this section, we evaluate the efficiency and the quality of our 3D-aware face editing model. We first introduce the implementation details of our work (Sec. 4.1). Subsequently, we compare our method with the SOTA 3D face editing methods qualitatively (Sec. 4.2) and quantitatively (Sec. 4.3). Then, we conduct ablation studies to analyze the effect of each component (Sec. 4.4). Finally, we introduce the potential applications of our method (Sec. 4.5).

4.1. Implementation Details

We build our approach on the EG3D [4] pre-trained on the FFHQ dataset [22]. We employ the Mediapipe [30] to detect 2D landmarks and select 29 points for user manipulation. To obtain 3D landmarks, we first detect 2D landmarks

Figure 3. Qualitative comparisons with current SOTA methods for 3D face shape and expression editing. (a), (b), and (c) are the results of synthetic samples, and (d) showcases the results of a real-world portrait.

in the frontal view, and then compute the 3D coordinates by the locations of maximum density value on their corresponding emitted rays. We adopt Swin-transformer [28] as the encoder structure to enhance the detail perception. In the encoder training, the standardized latent codes are sampled to generate the face images under random views, consisting of totally 100000 identities. We adopt the Adam optimizer [25] and set the learning rates as $1e-4$ for both the encoder and the discriminator. All the implementations are based on the PyTorch and set up on Nvidia A6000 GPUs.

4.2. Qualitative Evaluation

We conduct a qualitative comparison between our work and several SOTA 3D face editing methods with intuitive manipulation, *i.e.*, StyleGAN-NADA [13] guided by the text prompts and IDE-3D [46] controlled by the semantic maps. Besides, we also introduce the point-based warping approach into the qualitative comparison. We adopt similar editing objectives and use their official codes to ensure fairness. Fig. 3 shows the multi-view results of the shape and expression editing, demonstrating the superiority of our method on fine-grained modification. The warp can accomplish obvious editing, but it suffers from facial distortion. IDE-3D [46] achieve satisfied results in most cases. However, the coupling of different facial attributes in the semantic maps leads to changes beyond the target attributes. For instance, the baby in Fig. 3 (c) shows the shift of age and identity when trying to elongate his chin. Besides, IDE-3D only supports single-view editing, limiting its availability. StyleGAN-NADA [13] fails to edit the facial shape based on the EG3D despite its great success in style transfer and texture editing. In contrast, our method supports the user to simultaneously manipulate the face from multiple views and enables intuitive editing for facial shapes, expressions, and poses without the sacrifice of identity and detail. In addition to the editing quality, our method has another advantage that it does not require additional training for generative models, demonstrating its generalization.

Furthermore, we also compare our method with a recent 2D method, DragGAN [34], which employs a similar point-guided operation to ours. Since DragGAN is limited to 2D editing, we compare the results in two aspects, *i.e.*, fixed view editing and novel view synthesis, as shown in Fig. 4. In the aspect of fixed-view editing, the results of DragGAN [34] in Fig. 4 (a) show a tendency to open the mouth and change the identity when shortening the nose, although a mask limiting the editable region is applied. In the aspect of novel view synthesis, DragGAN severely changes the identity due to ambiguous point dragging in Fig. 4 (b). Compared to DragGAN, our method succeeds in achieving the expected editing target while maintaining the identity and irrelevant parts unchanged.

Methods	Scheme	Inference Time (s)↓	MSE_i↑	MSE_o↓	MSE_i / MSE_o↑	ID Consistency↑
DragGAN [34]	2D	5.231	1.992	0.224	8.893	0.579
Ours	2D	0.356	**2.049**	**0.186**	**11.016**	**0.716**
Our warp	3D	0.269	2.455	0.328	7.485	0.707
IDE-3D [46]	3D	**0.383**	**1.841**	0.987	1.865	0.649
Ours	3D	0.624	1.679	**0.342**	**4.909**	**0.712**

Table 2. Quantitative comparison with several face editing methods on efficiency and effectiveness. The best results are labeled in bold **except for** our direct warp due to its distortion results. The unit of MSE_i and MSE_o are 10^{-2}.

Figure 4. Qualitative comparisons with DragGAN [34] on portrait editing. Red and blue points represent the source and target points in the manipulations, respectively. The semi-transparent region indicates the mask used for DragGAN, while not in our method.

Figure 5. The ablation study of our loss functions for training the encoder. The first row aims to widen the double eyelids while keeping the eyes open, and the second is to lengthen the bangs. The numbers in the corners represent the identity similarity measured by ArcFace [10]. Please **zoom-in** for detailed observation.

4.3. Quantitative Evaluation

We also conduct quantitative experiments to verify the efficiency and effectiveness of our method, as shown in Tab. 2. We adopt editing time as the metric to evaluate the efficiency because it severely influences the user experiences. As shown, DragGAN [34] spends a large amount of time on latent optimization, resulting in lower efficiency. IDE-3D [46] and our method exhibit similar efficiency in supporting real-time editing. Despite the fastest method, the method of direct warp causes facial distortion, and thus we exclude it from the comparison.

Furthermore, to assess the capability of disentangled editing, we measure the pixel-wise mean square error (MSE) inside and outside the target editing regions as the metric. The main objective is to successfully edit the target regions while preventing the outside regions from modification. As shown, our approach achieves better editing disentanglement than IDE3D [46] with minimized ratio of MSE_i and MSE_o. It is worth noting that the editability of 3D GANs is inferior to that of 2D GANs, and thus our method falls behind the DragGAN [34]. Considering the efficiency and the ability to multi-view editing of our method, the gap between ours and the DragGAN is acceptable. To fairly compare these two methods without the interfere of base generators, we further extend our method to the same 2D generator and it performs better than DragGAN [34] in this setting. Additionally, we also compare the identity similarity. The results indicate that our method can better maintain the identity character than other methods.

4.4. Ablation Study

Effectiveness of Loss Functions. We investigate the effectiveness of each loss function in the encoder training process, as depicted in Fig. 5. The $\mathcal{L}_w$ introduced by the warp-assisted data augmentation facilitates the accurate identification for user's manipulations, and the $\mathcal{L}_d$ helps to maintain identity information. The combination of them achieves the best editing results.

Effectiveness of Directional Latent Editing. We conduct an ablation study to verify the effectiveness of our directional latent editing. We begin with applying tri-plane warping on source identities to obtain the warped results. Subsequently, we extract the directions of different layer groups, *i.e.*, shape direction, expression direction, and the combined directions, respectively. Fig. 6 shows that the individual directional latent code has the capacity to disentangle the attributes, while the combination of them can realize integrated editing. However, directly mapping warped rendering to latent space without our directional latent module results in identity shifting and detail deficiency. These results can verify the effectiveness of our directional latent editing.

4.5. Applications

Generalization of Learned Latent Directions. The editing direction learned for one face can generalized to other instances, and we can further control the degree along the

Figure 6. The ablation study of our directional editing. "w/o Dir." represents results generated by directly projecting the warped results to latent space.

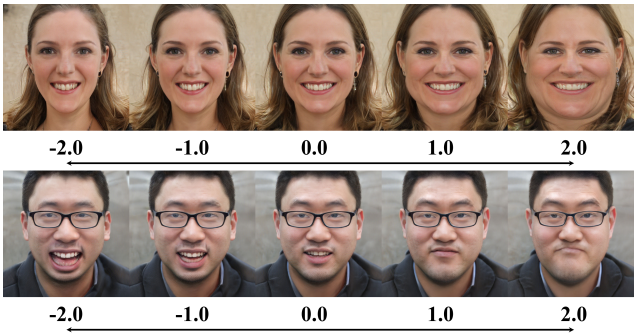


Figure 7. The interpolated editing results along the directions learned in the case of Fig. 3 (a) and (d), *i.e.*, "wider face" and "close mouth" respectively. It shows that the learned editing direction in one face can be generalized to other instances.

direction to linearly interpolate the editing results. Fig. 7 shows the interpolation results guided by the directions learned in the cases of Fig. 3, *i.e.*, wider face and closed mouth. With the degree rising from -2.0 to 2.0, both of the two identities show a gradual trend to change along their directions, although the directions are initially learned for other cases, demonstrating the generalization of these learned latent directions.

Continuous Editing. Continuous editing is important to real-world applications. Therefore, we conduct an experiment to show our capability of overlying modification. Fig. 8 shows the results with multiple editing targets, *i.e.*, smaller eyes, closed mouth, smaller nose, and wider face. The natural and ID-consistent results demonstrate the effectiveness of our method of continuous editing.

Generalization to Other Generators. To show the generalized application of our method, we extend it to 3D cat editing and 2D car editing. We introduce our method to the pre-trained EG3D [4] on AFHQ Cats [8] dataset and StyleGAN [23] trained on Stanford Cars [26] dataset, respectively. As shown in Fig. 9, our approach can also successfully manipulate the 3D cats and 2D cars according to the user's point-based instructions.

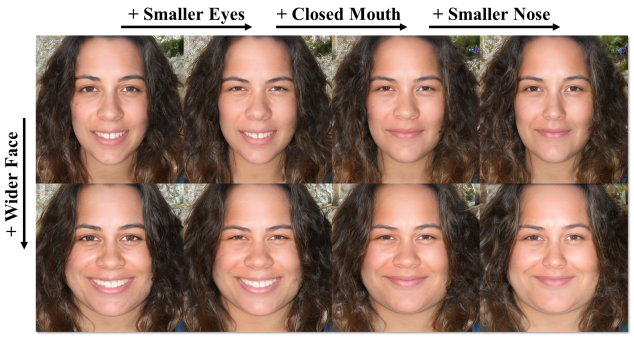
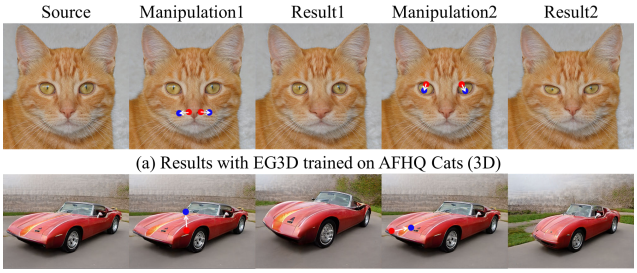


Figure 8. We showcase the mixing results with multiple attributes, demonstrating the continuous editing ability of our method.



(a) Results with EG3D trained on AFHQ Cats (3D)

(b) Results with StyleGAN trained on Stanford Cars (2D)

Figure 9. The extension of our method to cat and car editing.

5. Conclusion

In this paper, we propose FaceEdit3D, an intuitive method to edit the 3D facial shape and expression from any perspective. Our approach involves a tri-plane warping to ensure the inherent 3D-consistent editing. To mitigate facial distortions led by the warping, we train a warp-aware encoder to project the warped face into standardized distribution and further explore the hierarchical mechanism in latent space to achieve disentangled editing. Extensive experiments demonstrate the effectiveness and efficiency of our method. The additional applications also show the generalization and potential of our method across different applications. To sum up, our method provides a brand new way to manipulate the 3D representation, opening up new avenues for rapid and convenient real-image editing.

Limitations. Since our method is based on warping the 3D representation, it is hard for our work to achieve texture editing and some semantic editing, such as wearing glasses.

Broader Impacts. Despite not our intention, our 3D-aware facial editing capability could potentially be abused. We are committed to privacy protection, preventing the misuse of facial editing for criminal purposes.

Acknowledgements

This work was supported in part by NSFC (62201342, 62101325), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *TOG*, pages 1–21, 2021. 2, 3

[2] Sizhe An, Hongyi Xu, Yichun Shi, Guoxian Song, Umit Y Ogras, and Linjie Luo. Panohead: Geometry-aware 3d full-head synthesis in 360deg. In *CVPR*, pages 20950–20959, 2023. 1, 2, 3

[3] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021. 3

[4] 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, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022. 1, 2, 3, 5, 8

[5] Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *TOG*, pages 1–26, 2022. 3

[6] Zhuo Chen, Xudong Xu, Yichao Yan, Ye Pan, Wenhan Zhu, Wayne Wu, Bo Dai, and Xiaokang Yang. Hyperstyle3d: Text-guided 3d portrait stylization via hypernetworks. *arXiv preprint arXiv:2304.09463*, 2023. 2, 3

[7] Yuhao Cheng, Yichao Yan, Wenhan Zhu, Ye Pan, Bowen Pan, and Xiaokang Yang. Head3d: Complete 3d head generation via tri-plane feature distillation. *arXiv preprint arXiv:2303.15892*, 2023. 4

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020. 8

[9] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. Synthesizing normalized faces from facial identity features. In *CVPR*, pages 3703–3712, 2017. 4

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 5, 7

[11] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *CVPR*, pages 10673–10683, 2022. 1, 2

[12] Yuki Endo. User-controllable latent transformer for stylegan image layout editing. In *Computer Graphics Forum*, pages 395–406, 2022. 2

[13] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *TOG*, pages 1–13, 2022. 2, 3, 6

[14] Stephan J Garbin, Marek Kowalski, Virginia Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, Jingjing Shen, Matthew Johnson, and Julien Valentin. Voltemorph: Real-time, controllable and generalisable animation of volumetric representations. *arXiv preprint arXiv:2208.00949*, 2022. 3

[15] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2021. 1, 2, 3

[16] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *NeurIPS*, pages 9841–9850, 2020. 1, 2

[17] Clément Jambon, Bernhard Kerbl, Georgios Kopanas, Stavros Diolatzis, George Drettakis, and Thomas Leimkühler. Nerfshop: Interactive editing of neural radiance fields. *CGIT*, 6(1), 2023. 3

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

[19] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *ICCV*, pages 13799–13808, 2021. 3

[20] Wonjoon Jin, Nuri Ryu, Geonung Kim, Seung-Hwan Baek, and Sunghyun Cho. Dr. 3d: Adapting 3d gans to artistic drawings. In *SIGGRAPH Asia*, pages 1–8, 2022. 4

[21] Kacper Kania, Stephan J Garbin, Andrea Tagliasacchi, Virginia Estellers, Kwang Moo Yi, Julien Valentin, Tomasz Trzciński, and Marek Kowalski. Blendfields: Few-shot example-driven facial modeling. In *CVPR*, pages 404–415, 2023. 3

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 5

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 1, 8

[24] Gwanghyun Kim and Se Young Chun. Datid-3d: Diversity-preserved domain adaptation using text-to-image diffusion for 3d generative model. In *CVPR*, pages 14203–14213, 2023. 3, 4

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, pages 554–561, 2013. 8

[27] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *TOG*, pages 194–1, 2017. 3

[28] 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 *ICCV*, pages 10012–10022, 2021. 6

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 3

[30] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019. 5

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 99–106, 2020. 2

[32] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021. 1, 2, 3

[33] 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 *CVPR*, pages 13503–13513, 2022. 1, 2, 3

[34] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ASIGGRAPH*, pages 1–11, 2023. 2, 3, 6, 7

[35] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5865–5874, 2021. 3

[36] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *CVPR*, pages 2085–2094, 2021. 1, 2, 3, 5

[37] 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. *NeurIPS*, pages 31402–31415, 2022. 3

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 2

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[40] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *CVPR*, pages 2886–2897, 2021. 3

[41] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NIPS*, 2020. 1, 2, 3

[42] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, pages 1532–1540, 2021. 2, 3

[43] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI*, pages 2004–2018, 2020. 1, 2, 3

[44] Enis Simsar, Alessio Tonioni, Evin Pinar Ornek, and Federico Tombari. Latentswap3d: Semantic edits on 3d image gans. In *ICCV*, pages 2899–2909, 2023. 2

[45] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *NeurIPS*, pages 24487–24501, 2022. 1, 2, 3

[46] 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. *ToG*, pages 1–10, 2022. 1, 2, 3, 6, 7

[47] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *CVPR*, pages 7672–7682, 2022. 3

[48] Jingxiang Sun, Xuan Wang, Lizhen Wang, Xiaoyu Li, Yong Zhang, Hongwen Zhang, and Yebin Liu. Next3d: Generative neural texture rasterization for 3d-aware head avatars. In *CVPR*, pages 20991–21002, 2023. 1, 2

[49] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6142–6151, 2020. 2

[50] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, pages 9786–9796, 2020. 2, 3

[51] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. In *CVPR*, pages 4563–4573, 2023. 1

[52] Sijing Wu, Yichao Yan, Yunhao Li, Yuhao Cheng, Wenhan Zhu, Ke Gao, Xiaobo Li, and Guangtao Zhai. Ganhead: Towards generative animatable neural head avatars. In *CVPR*, pages 437–447, 2023. 3

[53] Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds. In *ICCV*, pages 2195–2205, 2023. 1, 2, 3

[54] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *ECCV*, pages 159–175, 2022. 3

[55] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, 2022. 1, 2

[56] Yan Yichao, Cheng Yuhao, Chen Zhuo, Peng Yicong, Wu Sijing, Zhang Weitian, Li Junjie, Li Yixuan, Gao Jingnan, Zhang Weixia, Zhai Guangtao, and Yang Xiaokang. A survey on generative 3d digital humans based on neural networks: representation, rendering, and learning. *SCIENTIA SINICA Informationis*, pages 1858–, 2023. 1

[57] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *CVPR*, pages 18353–18364, 2022. 3

[58] 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. In *ICCV*, pages 2437–2447, 2023. 5

[59] Chi Zhang, Yiwen Chen, Yijun Fu, Zhenglin Zhou, Gang Yu, Billzb Wang, Bin Fu, Tao Chen, Guosheng Lin, and Chunhua Shen. Styleavatar3d: Leveraging image-text diffusion models for high-fidelity 3d avatar generation. *arXiv preprint arXiv:2305.19012*, 2023. 2

[60] Jianfeng Zhang, Zihang Jiang, Dingdong Yang, Hongyi Xu, Yichun Shi, Guoxian Song, Zhongcong Xu, Xinchao Wang,

and Jiashi Feng. Avatargen: a 3d generative model for animatable human avatars. In *ECCV*, pages 668–685. Springer, 2022. 5

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 5

[62] Xuanmeng Zhang, Jianfeng Zhang, Rohan Chacko, Hongyi Xu, Guoxian Song, Yi Yang, and Jiashi Feng. Getavatar: Generative textured meshes for animatable human avatars. In *ICCV*, pages 2273–2282, 2023. 5

[63] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. Im avatar: Implicit morphable head avatars from videos. In *CVPR*, pages 13545–13555, 2022. 3

[64] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Cips-3d: A 3d-aware generator of gans based on conditionally-independent pixel synthesis. *arXiv preprint arXiv:2110.09788*, 2021. 1, 2, 3

[65] Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zheng-Jun Zha, Jingren Zhou, and Qifeng Chen. Low-rank subspaces in gans. *NeurIPS*, pages 16648–16658, 2021. 2

[66] Jiapeng Zhu, Yujun Shen, Yinghao Xu, Deli Zhao, and Qifeng Chen. Region-based semantic factorization in gans. In *ICML*, pages 27612–27632, 2022.

[67] Jiapeng Zhu, Ceyuan Yang, Yujun Shen, Zifan Shi, Bo Dai, Deli Zhao, and Qifeng Chen. Linkgan: Linking gan latents to pixels for controllable image synthesis. In *ICCV*, pages 7656–7666, 2023. 2

[68] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, pages 5104–5113, 2020. 3

[69] Peiye Zhuang, Oluwasanmi Koyejo, and Alexander G Schwing. Enjoy your editing: Controllable gans for image editing via latent space navigation. *arXiv preprint arXiv:2102.01187*, 2021. 3