

Training and Tuning Generative Neural Radiance Fields for Attribute-Conditional 3D-Aware Face Generation

JICHAO ZHANG, University of Trento, Italy
 ALIAKSANDR SIAROHIN, Snap Research, US
 YAHUI LIU, University of Trento, Italy
 HAO TANG, ETH Zurich, Switzerland
 NICU SEBE, University of Trento, Italy
 WEI WANG, University of Trento, Italy

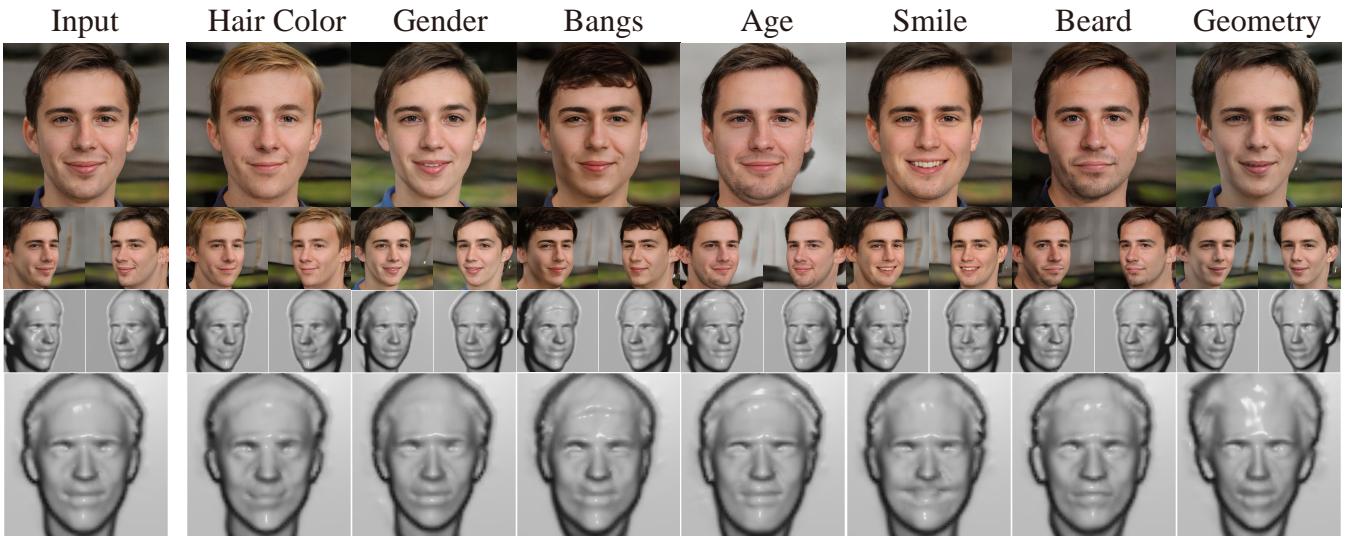


Fig. 1. Controllable 3D-aware face generation (first two rows) produced by our method given specific attributes as guidance, and the corresponding mesh (bottom two rows). As shown in the meshes, we can observe that the geometry has been preserved for the attribute “Hair Color” and “Bangs”, while the mouth region of the “Smiling” mesh has been changed, moreover the shape of the mesh with altered “Geometry” has been enlarged in size.

3D-aware GANs based on generative neural radiance fields (GNeRF) have achieved impressive high-quality image generation, while preserving strong 3D consistency. The most notable achievements are made in the face generation domain. However, most of these models focus on improving view consistency but neglect a disentanglement aspect, thus these models cannot provide high-quality semantic/attribute control over generation. To this end, we introduce a conditional GNeRF model that uses specific attribute labels as input in order to improve the controllabilities and disentangling abilities of 3D-aware generative models. We utilize the pre-trained 3D-aware model as the basis and integrate a dual-branches attribute-editing module (DAEM), that utilize attribute labels to provide control over generation. Moreover, we propose a TRIOT (TRaining as Init, and Optimizing for Tuning) method to optimize the latent vector to improve the precision of the attribute-editing further. Extensive experiments on the widely used FFHQ show that our model yields high-quality editing with better view consistency while preserving the non-target regions. The code is available at <https://github.com/zhangqianhui/TT-GNeRF>.

Authors’ addresses: Jichao Zhang, jichao.zhang@unitn.it, University of Trento, Trento, Italy; Aliaksandr Siarohin, Snap Research, CA, US, aliaxsandr.siarohin@snap.com; Yahui Liu, University of Trento, Trento, Italy, yahui.liu@unitn.it; Hao Tang, ETH Zurich, Zurich, Switzerland, hao.tang@vision.ee.ethz.ch; Nicu Sebe, University of Trento, Trento, Italy, sebe@disi.unitn.it; Wei Wang, University of Trento, Trento, Italy, wangwei1990@gmail.com.

CCS Concepts: • Computing methodologies → Computer vision tasks; Image manipulation.

Additional Key Words and Phrases: Neural radiance fields, Generative Adversarial Networks, 3D-Aware Face Generation and Editing

1 INTRODUCTION

High-quality image generation and semantic disentanglement are the longstanding goals of computer vision and computer graphics. In recent years, Generative Adversarial Networks (GANs) [Goodfellow et al. 2014], and their variants (e.g., StyleGAN [Karras et al. 2021, 2019, 2020]) have been drawing booming attention and opening the world for high-quality image generation and editing. These methods significantly improve the visual fidelity, speed of the image rendering, and interactive controls compared to the traditional computer graphics pipelines.

Many prior works [Abdal et al. 2021; Choi et al. 2018; Shi et al. 2021; Tewari et al. 2020] focus on realistic face editing. They either directly base on an image-to-image translation models [Choi et al. 2018; Liu et al. 2019], or utilize the disentangling abilities [Abdal et al. 2021; Shi et al. 2021; Tewari et al. 2020] of StyleGAN [Karras et al.

2019, 2020]. We can briefly classify them into supervised and unsupervised categories. Regarding the unsupervised methods, most of them search the interpretable directions using PCA [Härkönen et al. 2020], or introducing orthogonalization [He et al. 2021; Voynov and Babenko 2020] in the latent space. However, these methods can provide only coarse controls. Thus, supervised methods [Abdal et al. 2021; Choi et al. 2018] utilize the specific attribute labels as a condition. However, they lack precise controls on 3D factors such as camera pose because they are inclined to ignore the underlying 3D scene rendering process. To alleviate this problem, some works [Deng et al. 2020; Tewari et al. 2020] integrate 3D Morphable Face Models (3DMM) [Paysan et al. 2009] to provide controls on 3D face pose and facial expression. However, these works still suffer from serious problems like view-inconsistency and unrealistic texture distortion when we drastically vary the poses.

Recently, Neural radiance fields (NeRF) [Mildenhall et al. 2020] have attracted booming attention because of their impressive results in novel view-rendering tasks. Specifically, NeRF represents a scene using a continuous function, parameterized by a multi-layer perceptron (MLP) that maps a 3D position and a viewing direction to density and radiance values. Since then, many works have been proposed to improve NeRF [Müller et al. 2022; Zhang et al. 2020] and apply it to various downstream tasks, such as human body modeling [Peng et al. 2021b] and large scene modeling [Tancik et al. 2022].

Some 3D-aware image generation methods [Chan et al. 2021; Yen-Chen et al. 2021] combine NeRF with the generative model by extending the neural radiance fields with latent conditioning, referred to as Generative Neural Radiance Fields (GNeRF). In detail, the 3D coordinates are sampled from random camera poses and then used as input to an implicit function along with additional latent codes. As in regular NeRF, this function is used to predict density and RGB color. However, these methods are compute-intensive and memory inefficient since they require lots of sample points in each ray. Thus, they are limited to low-resolution and low-quality generation. To address this issue, GIRAFFE [Niemeyer and Geiger 2021] learns to generate low-resolution feature fields and use a convolution-based neural rendering module to map the rendered features into the high-resolution output. However, it suffers from serious view-inconsistency problems. To improve generation quality and view-consistency, lots of approaches [Chan et al. 2022; Or-El et al. 2022; Zhang et al. 2022] borrow ideas from StyleGAN and integrate the ‘Style-modules’ into the implicit function (e.g., SIREN [Chan et al. 2021]) or neural rendering module. Additionally, some novel algorithms and losses have been elaborately designed for 3d-aware generation, such as tri-planes [Chan et al. 2022] or multiple-view warping loss [Zhang et al. 2022]. While these models create high-quality and view-consistent images, they lack the control and disentangling abilities. As explained in VolumeGAN [Xu et al. 2022a], some models are limited to local receptive fields with MLPs, and it is hard to extract global structures from their internal representation. Thus, VolumeGAN utilizes a 3D feature volume module for querying coordinate descriptors, enabling independent controls on the texture and structure factors. However, VolumeGAN is still limited to the vital quality and view-consistency problems.

Moreover, it does not support attribute controls for face manipulation.

We introduce an attribute-conditional 3D-aware generative model for controlling facial attributes to solve the problems mentioned above. Compared to the NeRF-based Head-Avatar models [Hong et al. 2022; Zhuang et al. 2021], there are two noticeable differences. First, our model is free of 3DMM priors that do not need to model the coefficients of 3DMM. Second, we utilize a pretrained 3D-aware GNeRF as the backbone to avoid retraining the entire model. Notably, we propose a dual-branches attribute editing module (DAEM) to edit the latent code in a specific interpretable direction (See Fig. 2). To train the DAEM module, we fix the pretrained GNeRF and train only the DAEM module by sampling the training triplets: the latent codes from the GNeRF, the corresponding generated face images, and the labels obtained by the classifiers. Note that a similar triplet sampling strategy was also utilized in StyleFlow [Abdal et al. 2021] which learns to edit the latent space for 2D-GAN models. To further improve the results and better preserve non-target regions of the image, we propose a novel “Training-as-Init, Optimizing-for-Tuning” method (TRIOT) (See the left of Fig. 3). In the proposed TRIOT, we use the edited latent vector for target attributes as initialization and then optimize this latent vector with the proposed semantic-guided texture and geometry consistency losses while fixing the rest of the model. Finally, we present an unsupervised optimization method for editing the geometry of the face (See the right of Fig. 3).

In summary, the main contributions of this work are:

- 1) We propose a dual branches attribute-editing module (DAEM) to promote the controllability and disentanglement of the 3D-aware generative model.
- 2) We present a novel learning method, ‘Training-as-Init, Optimizing-for-Tuning’ (TRIOT), combining the model-training and latent-optimization method for the attribute-editing task.
- 3) We show that, compared to the baselines, our model can achieve high-quality editing with better view consistency while preserving the non-target regions.
- 4) We propose an unsupervised optimization method for editing the geometry of faces.

2 RELATED WORK

Generative Neural Radiance Fields for 3D-Aware Face Generation. Neural radiance fields (NeRF) [Mildenhall et al. 2020] is a continuous neural mapping from a 3D position and a 2D viewing direction to the RGB value and density that allows 3D scene modeling and high-quality novel view synthesis. Recently, several NeRF-based methods have been proposed to improve rendering speed [Barron et al. 2021; Müller et al. 2022; Reiser et al. 2021] and rendering quality [Barron et al. 2022; Niemeyer et al. 2022; Verbin et al. 2022]. Moreover, NeRF also promotes the development of many computer-graphics applications, such as human body modeling [Peng et al. 2021a,b], 3D-aware face generation [Chan et al. 2022; Gu et al. 2022; Ma et al. 2022; Niemeyer and Geiger 2021; Or-El et al. 2022; Xu et al. 2022a], large scene modeling [Tancik et al. 2022], and pose estimation [Yen-Chen et al. 2021].

Generative neural radiance fields (GNeRF) are a conditional variant of NeRF, which combines NeRF with GAN to condition the

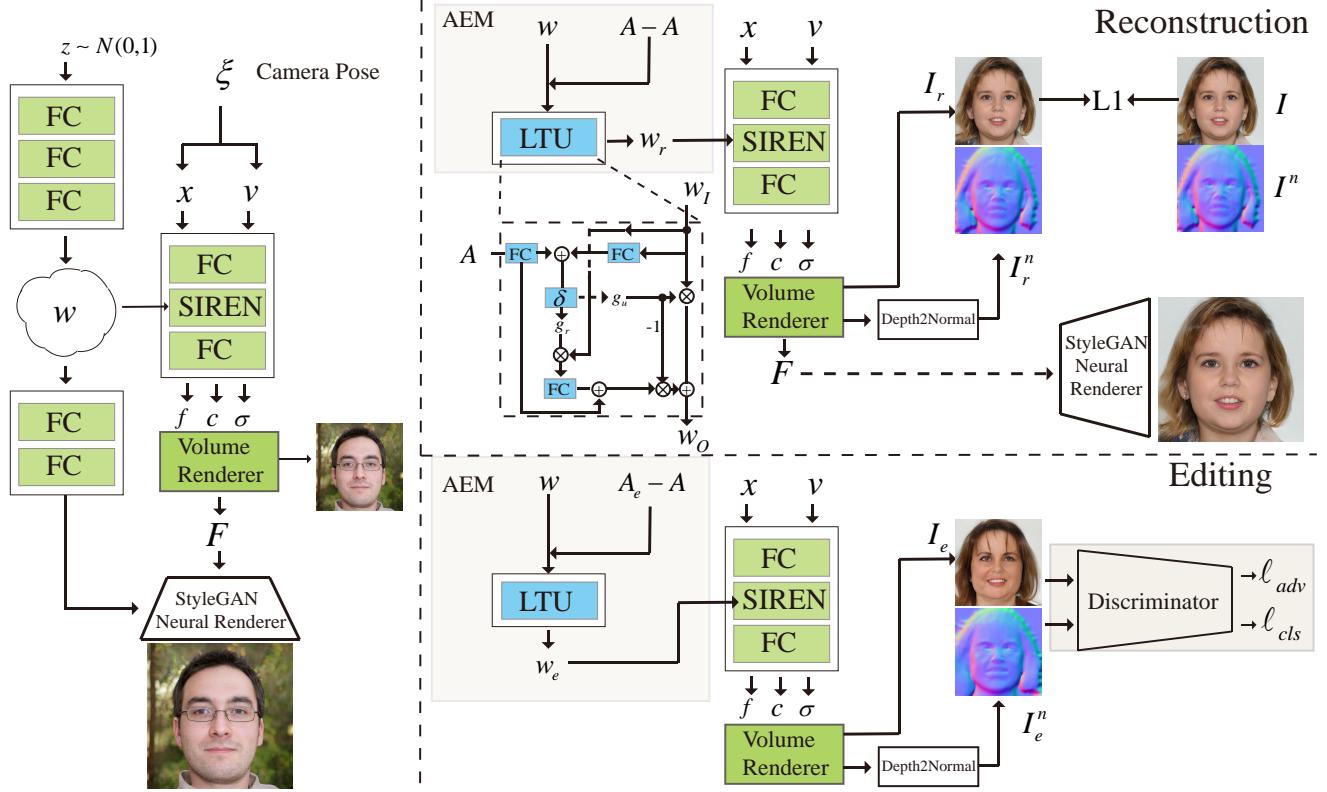


Fig. 2. The architecture of our backbone StyleSDF (Left) [Or-El et al. 2022]. The overview of the pretrained StyleSDF with double branches attribute-editing module (DAEM) (Right). One branch (AEM) performs the reconstruction using the label residual $A - A = 0$ as the input, and the other branch performs the editing using the label residual $A_e - A$ as the input.

rendering process on a latent code that govern the object’s appearance and shape [Chan et al. 2021; Niemeyer and Geiger 2021; Yen-Chen et al. 2021]. For example, GRAF [Schwarz et al. 2020] achieves this goal by incorporating shape and appearance codes as input. GRAF [Schwarz et al. 2020] achieves better visual fidelity and view consistency than the previous voxel- and feature-based methods [Henzler et al. 2019; Nguyen-Phuoc et al. 2019]. Michael et al. [Niemeyer and Geiger 2021] propose the compositional neural feature fields (GIRAFFE) that extend GRAF into 3D-aware multiple-object scene representations. Although GRAF and GIRAFFE can control texture and camera pose, they are limited to low-resolution results and fail to preserve multi-view consistency. Many works [Chan et al. 2021, 2022; Deng et al. 2022; Gu et al. 2022; Or-El et al. 2022; Pan et al. 2021; Schwarz et al. 2022; Skorokhodov et al. 2022; Xiang et al. 2022; Xue et al. 2022; Zhang et al. 2022; Zhou et al. 2021b] are trying to address these problems, and most of them inherit the “image-as-style” idea from StyleGAN [Karras et al. 2020]. Yang et al. [Xue et al. 2022] extend the GIRAFFE to work with high-resolution data. However, this model still suffers from the view-inconsistency problems. Pi-GAN [Chan et al. 2021] presents a SIREN module with periodic activation functions. It conditions the style code through feature-wise linear modulation (FiLM). The SIREN modules significantly boost image quality and view consistency. To reduce the high computational costs of the volume rendering in Pi-GAN, some models, such as StyleSDF [Or-El et al. 2022], MVCGAN [Zhang

et al. 2022] and EG3D [Chan et al. 2022] propose a hybrid rendering approach. Specifically, they learn a coarse feature field, render it into a low-resolution feature map, and then utilize a style-based 2D network as a “super-resolution” module to refine the features for a final high-resolution image. In order to improve view consistency, StyleSDF models signed distance fields, while MVCGAN uses the explicit multi-view consistency loss. On the other hand, ED3D proposes a hybrid 3D tri-plane representation. Unlike the mentioned works, CIPS-3D [Zhou et al. 2021b] keeps the resolution of the intermediate feature fields the same as the resolution of the final images. Though these models can achieve incredible quality generation with strong view-consistency, they cannot edit structures and textures.

Recently, some research has focused on the disentangling abilities of the 3D-aware models. VolumeGAN [Xu et al. 2022a] tries to separate shape from texture, while ShadeGAN [Pan et al. 2021] disentangles the light from the albedo. However, they only focus on the global factors, such as illumination and textures, and cannot handle more specific attributes, such as hair color and gender. To address this problem, we propose an attribute-conditional 3D-aware GAN model that inputs specific attribute labels. Compared to the methods exploiting 3DMM prior [Hong et al. 2022; Zhuang et al. 2021], we propose to leverage the pretrained GNeRF and integrate a double-branches attribute-editing module (DAEM), which takes

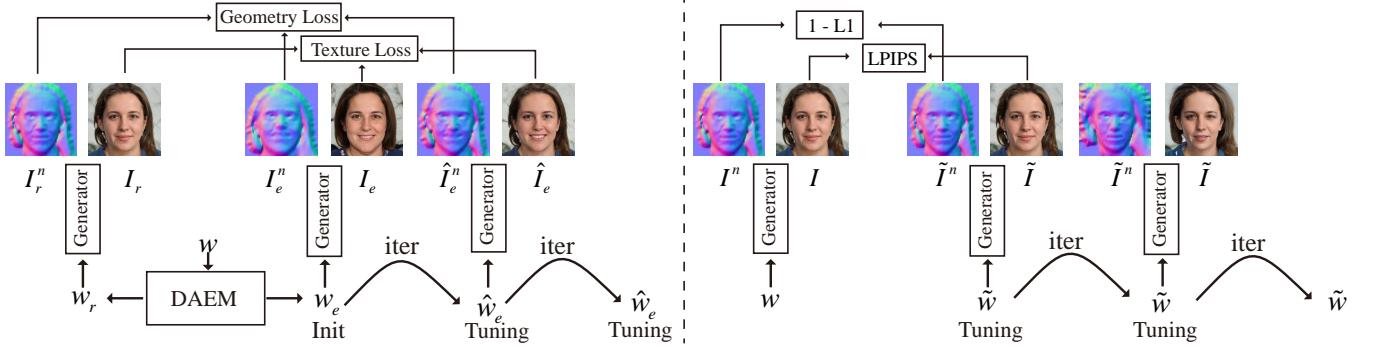


Fig. 3. Left: The pipeline of TRIOT with geometry consistency loss and texture consistency loss for the smiling editing. Right: the geometry editing optimization pipeline maximizes the differences between normal images and minimizes the differences in perceptual space.

specific attributes as input. Moreover, we propose a novel “Training-as-init, Optimized-for-Turning” method (TRIOT) to train and optimize attribute-editing modules (AEM), that helps to achieve better 3D-aware face generation and editing while preserving non-target regions.

Image-to-Image Translation Architectures for Face Editing. Image-to-Image translation models, *i.e.*, Pix2Pix [Isola et al. 2017] and CycleGAN [Zhu et al. 2017], utilize the autoencoder as generator that have been widely adopted for a variety of different tasks, including face attribute editing [Chen et al. 2020; Choi et al. 2018; Chu et al. 2020; Gao et al. 2021; He et al. 2020, 2019; Liu et al. 2019, 2021b; Wu et al. 2019]. Specifically, StarGAN [Choi et al. 2018] is the early work for learning multiple-domain face translation, it takes multiple attributes as input, and can transfer one face image from one domain to other domains. During the training, StarGAN exploits a reconstruction and cycle-consistency loss to preserve the content of the input face. After that, many works have been improving StarGAN, such as AttGAN [He et al. 2019], STGAN [Liu et al. 2019], SSCGAN [Chu et al. 2020] and HifaFace [Gao et al. 2021]. However, all these models are limited to low-resolution data and 2D face editing, *i.e.*, they cannot manipulate 3D factors of the face, such as camera poses.

Interpreting Latent Space of StyleGAN for Face Editing Alternative line of works explore the disentanglement of latent space of StyleGAN for face editing. These approaches can be roughly classified into two types, according to the usage of the semantic labels, *i.e.*, unsupervised methods and attributes-conditional methods.

The former learns to discover interpretable directions in latent space by leveraging Principal Component Analysis (PCA) [Härkönen et al. 2020] (*e.g.*, using closed-form factorization [Shen and Zhou 2021]) by utilizing a learnable orthogonal matrix [He et al. 2021; Voynov and Babenko 2020] or by applying the regularization losses [Peebles et al. 2020; Wei et al. 2021]. GANSpace [Härkönen et al. 2020] shows that PCA in the latent space of StyleGAN can find important interpretable directions that can be utilized to control image generation. In order to compute the PCA of the style codes, GANSpace samples multiple random vectors (*i.e.*, z space) and computes the corresponding style codes (*i.e.*, \mathcal{W} space). To avoid the extensive data sampling of GANSpace, SeFa [Shen and Zhou 2021] decomposes the model weights directly with a closed-form solution. Similarly, recent works [He et al. 2021; Voynov and Babenko

2020] propose to obtain a disentangled latent space by learning an orthogonal matrix for editing latent code.

As far as we know, the attribute conditions can be of different types, including global-level (*e.g.*, label vectors) and local-level (*e.g.*, semantic segmentation maps) modalities. The first type [Abdal et al. 2021; Liang et al. 2021; Liu et al. 2021a] usually utilizes the off-the-shelf attribute classifier networks to obtain the attribute vectors of the training images and then uses these vectors as input. For example, StyleFlow [Abdal et al. 2021] propose to utilize conditional normalizing flow (CNF) to model the mapping from the conditional labels and latent codes (z space) to intermediate vectors (\mathcal{W} space). StyleFlow trains the flow model (CNF) with triplets, consisting of vectors sampled from \mathcal{W} space, corresponding faces and predicted face attributes. Though StyleFlow can produce facial pose transformation, it suffers from serious view-inconsistency, as it lacks understanding of the underlining 3D world. The second type utilizes the coarse masks [Zhu et al. 2022] or predict face semantics [Collins et al. 2020] with k-means.

Additionally, some methods [Kim et al. 2021; Kwon and Ye 2021; Shi et al. 2022; Xu et al. 2022b] also explore the semantic disentanglement of the model, but they redesign the StyleGAN, thus they need to retrain the generator. For example, TransEditor [Xu et al. 2022b] presents a transformer-based module for dual space interactions, where one latent code is used as the key and value and the other as the query. This helps the disentanglement of the style and the content representations. Some works focus on local facial controls by integrating face parsing into generation [Shi et al. 2022] or by adding spatial information for styles code with the conv-based module [Kim et al. 2021].

Overall, these StyleGAN-based models have demonstrated the ability to produce high-quality images and perform precise editing. However, they fail to change the facial pose and preserve view consistency, owing to a lack of 3D modeling abilities.

3DMM-Guided Face Generation and Editing. Recently, some works [Deng et al. 2020; Geng et al. 2019; Lin et al. 2022; Shi et al. 2021; Tewari et al. 2020] demonstrate high-quality control over GAN generation via a 3DMM [Paysan et al. 2009]. 3DMM is the 3D Morphable Face Model, parameterized by the face shape, expression, and texture. For example, Geng *et al.* [Geng et al. 2019] utilizes 3DMM to guide fine-grained face manipulation for arbitrary expression transfer. First, they extract texture and shape coefficients

by fitting 3DMM to each real face in the dataset. Then they utilize the texture generator to create the target textures with the source texture and the target expression and utilize the shape predictor to produce the target shape with the source shape coefficients and the target expression as input. Finally, the global generator utilizes rendered faces and the target expression to produce the final faces. StyleRig [Tewari et al. 2020] and DiscoFaceGAN [Deng et al. 2020] use 3DMM to manipulate the latent space of StyleGAN. While StyleRig is based on pretrained StyleGAN models and only tunes a DFR module that learns the mapping from latent code to the coefficients of 3DMM. On the other hand, DiscoFaceGAN re-trains the entire model. It exploits multiple VAE to model the distribution of 3DMM coefficients and introduces self-supervised losses to disentangle different factors. Compared to these models, our model does not require a 3DMM prior and still achieves better multi-view consistency. Additionally, our models can achieve more variable face editing, such as hair color and age.

Finally, Shi *et al.* [Shi et al. 2021] presents a LiftedGAN model, which lifts the pretrained StyleGAN2 in 3D. This model is free of 3DMM prior. However this model cannot achieve attribute-conditional control.

GAN Inversion for Real Face Editing. GAN inversion aims to find an optimal latent code corresponding to the given real image. It has been widely used for real image editing tasks. The previous methods can be divided into two broad categories: optimization-based [Abdal et al. 2019, 2020; Roich et al. 2021] and encoder-based [Alaluf et al. 2021; Richardson et al. 2021; Tov et al. 2021; Zhu et al. 2020]. For example, Roich *et al.* [Roich et al. 2021] presents a novel optimization-based method called Pivotal Tuning Inversion (PTI). In PTI, they first obtain the optimized latent code as the pivot by fixing the parameters of the generator, then fix this pivot and fine-tune the generator parameters to obtain better reconstruction while preserving the editing abilities of the latent code. After the inversion step, they utilize the popular latent-disentanglement method, such as InterfaceGAN or GANSpace, for face editing. In this paper, we use the PTI for GAN inversion. Concurrent with our work, some methods [Chen et al. 2022; Sun et al. 2022a,b] employ 3D-aware GAN as the basic model instead of StyleGAN to achieve multi-view consistent face editing, guided by the segmentation masks. They also apply GAN inversion to project real images into the latent space for editing. Different from these methods with the segmentation masks, we use attribute labels to guide face editing.

3 METHODS

We start from the introduction of the 3D-aware GAN with generative neural radiance fields. Our method can potentially work with most of GNeRF backbones, and we showcase it with the two most recent ones StyleSDF [Or-El et al. 2022] and EG3D [Chan et al. 2022]. Since they have similar architecture, we only describe StyleSDF [Or-El et al. 2022]. Then, we detail the proposed double branches attribute editing module (DAEM) and ‘Training-As-Init, Optimizing-for-Tuning’ (TRIOT) method with attribute-specific consistency loss. Finally, we introduce our unsupervised optimization method for facial geometry editing.

3.1 Generative Neural Radiance Fields (GNeRF)

NeRF [Mildenhall et al. 2020] is a continuous neural mapping M , which maps a 3D position \mathbf{x} and a 2D viewing direction \mathbf{v} to the rgb color \mathbf{c} and the density σ :

$$(\mathbf{c}, \sigma) = M(\gamma(\mathbf{x}), \gamma(\mathbf{v})), \quad (1)$$

where γ indicates the positional encoding mapping function.

GNeRF [Schwarz et al. 2020] is a conditional variant of NeRF. Unlike NeRF, it requires multiple views of a single scene with estimated camera poses. Notably, GNeRF can be trained with unposed 2D images from different scenes. In Pi-GAN [Chan et al. 2021], GNeRF is trained with adversarial learning, and it is conditioned on a latent code z :

$$(\mathbf{c}, \sigma) = M(\gamma(\mathbf{x}), \gamma(\mathbf{v}), \mathbf{z}), \quad (2)$$

where the latent code z with the following MLP layers aims to infer the frequencies α and the shifts β of a SIREN layer [Chan et al. 2021].

StyleSDF. As shown in Fig. 2, StyleSDF also adopts the SIREN layers inside GNeRF. However, it utilizes Signed Distance Fields (SDF) to improve the GNeRF and add a 2D StyleGAN generator as a second stage rendering. In the first stage, the GNeRF is trained separately. It produces a feature vector \mathbf{f} , RGB color \mathbf{c} and SDF values \mathbf{d} :

$$(\mathbf{f}, \mathbf{c}, \mathbf{d}) = M(\gamma(\mathbf{x}), \gamma(\mathbf{v}), \mathbf{w}), \quad (3)$$

where the learned SDF values define the object surface and thus allow to extract of the mesh via Marching Cubes [Lorensen and Cline 1987]. Moreover, \mathbf{d} will be converted into the density σ for volume rendering.

The RGB color \mathbf{c} is later rendered into the low-resolution face image, following the classical volume rendering. Then, the discriminator takes the output image as an input for adversarial training.

In the second stage, all parameters of the GNeRF are fixed. The feature vector \mathbf{f} is volume-rendered into low-resolution feature map \mathbf{F} , and \mathbf{F} is mapped into a high-resolution result using the Style-based convolutional modules. This high-resolution image is passed to another discriminator.

3.2 Double Branches Attribute Editing Module (DAEM)

As mentioned before, we use the pre-trained StyleSDF (or EG3D) as the backbone of our model. First, we sample training triplets from the pretrained StyleSDF model: latent vector \mathbf{w} , the corresponding generated sample \mathbf{I} along with its low-resolution \mathbf{I}_L version and attribute labels \mathbf{A} predicted by the *off-the-shelf* attribute classifiers. Then, we extend the pretrained StyleSDF with the proposed Double-Branch Attribute Editing Module (DAEM), which manipulates the latent code for 3D-aware attribute editing. Our DAEM consists of a reconstruction branch and an editing branch, while every branch contains an Attribute Editing Module (AEM). For the AEM, we propose to utilize the Latent Transfer Unit (LTU) blocks, which is a variant of the Gated Recurrent Unit (GRU) [Chung et al. 2014], in order to improve the quality of the transfer.

Latent Transfer Unit (LTU). In general, facial attribute editing has an editing-preservation trade-off. To achieve accurate editing of the target region while preserving the non-target regions, we need to manipulate some dimensions of the latent code while keeping the remaining dimensions unchanged. Thus, we follow the idea of the

Gated Recurrent Unit (GRU), which has the reset and update gates to control how much the unit forgets and updates the information. Our Latent Transfer Unit (LTU) is specifically designed for modeling editing and preservation of the different latent code dimensions.

As shown in the right of Fig. 2, we take the label \mathbf{A} as a condition for our LTU and one latent code \mathbf{w}_I as the latent input. First, we obtain the values of the gate:

$$\begin{aligned}\mathbf{G}_r &= \sigma(FC(FC(\mathbf{A}) + FC(\mathbf{w}_I))) \\ \mathbf{G}_u &= \sigma(FC(FC(\mathbf{A}) + FC(\mathbf{w}_I))),\end{aligned}\quad (4)$$

where \mathbf{G}_r and \mathbf{G}_u are reset gate and update gate, σ is the Sigmoid function, and FC is some fully-connected layers. Then, the candidate latent with new information can be defined as:

$$\tilde{\mathbf{w}}_I = FC(FC(\mathbf{A}) + FC(\mathbf{G}_r \odot \mathbf{w}_I)). \quad (5)$$

Finally, we can obtain the updated latent code:

$$\mathbf{w}_O = \mathbf{G}_u \odot \tilde{\mathbf{w}}_I + (1 - \mathbf{G}_u) \odot \mathbf{w}_I, \quad (6)$$

where we do not use Tanh function for $\tilde{\mathbf{w}}_I$ to preserve the same range of value between $\tilde{\mathbf{w}}_I$ and \mathbf{w}_I , compared to the typical GRU.

In the following part, we introduce the details of our reconstruction and editing branches.

Reconstruction branch. As shown in the right of Fig. 2, our single attribute-editing module (AEM) AEM_r consists of a one-layer LTU that takes the latent code \mathbf{w} and the label residual $\mathbf{A} - \mathbf{A} = 0$ as the input, and output new latent vector \mathbf{w}_r :

$$\mathbf{w}_r = AEM_r(\mathbf{w}, \mathbf{A} - \mathbf{A}). \quad (7)$$

The learned \mathbf{w}_r is mapped into the frequencies α and the shifts β by several fully-connected layers and used as an input of the SIREN module. After that, we can obtain the reconstruction face image I_r and the normal map I_r^n with pretrained StyleSDF. Note that we convert the density into the depth, and later we convert the depth into the normal map by using the cross product of neighboring pixels to represent the geometry.

Editing branch. Similar to the reconstruction branch, our editing branch AEM_e is:

$$\mathbf{w}_e = AEM_e(\mathbf{w}, \mathbf{A}_e - \mathbf{A}), \quad (8)$$

where \mathbf{A}_e is a random target label. With the pretrained StyleSDF, we can obtain an editing face image I_e and a normal map I_e^n .

Loss Functions. For training DAEM, our overall objective function consists of three items: adversarial loss \mathcal{L}_{adv} , classification loss \mathcal{L}_{cls} and reconstruction loss \mathcal{L}_{recon} , and it is defined as:

$$\mathcal{L}_{DAEM} = \mathcal{L}_{adv} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{recon}. \quad (9)$$

where λ_1 and λ_2 are hyper-parameters that control the contribution of the corresponding loss terms.

Different from other 3D-aware models [Or-El et al. 2022; Zhou et al. 2021b], our discriminator D takes the image I and normal map I^n as input to learn consistent texture and geometry. The adversarial loss \mathcal{L}_{adv} is defined as:

$$\begin{aligned}\min_{DAEM} \max_D \mathcal{L}_{adv} &= \mathbb{E}_{I, I^n} [\log D_{adv}(I, I^n)] \\ &+ \mathbb{E}_{I_e, I_e^n} [\log(1 - D_{adv}(I_e, I_e^n))].\end{aligned}\quad (10)$$

The classification loss \mathcal{L}_{cls} is defined as:

$$\begin{aligned}\mathcal{L}_{cls} &= \mathbb{E}_{I, \mathbf{A}} [-\log D_{cls}(\mathbf{A}|I)] \\ &+ \mathbb{E}_{I_e, \mathbf{A}_e} [-\log(D_{cls}(\mathbf{A}_e|I_e))].\end{aligned}\quad (11)$$

Finally, the reconstruction loss \mathcal{L}_{recon} is defined as:

$$\mathcal{L}_{recon} = \|I_r - I\|_1 + \|\mathbf{w}_r - \mathbf{w}\|_1. \quad (12)$$

Note that reconstruction loss only guides the reconstructed image I_r , while the adversarial and classification loss guides only the edited image I_e .

3.3 Training-as-Init, Optimizing-for-Tuning with Attribute-Specific Consistency objection

After training, our DAEM is with the ability to edit the attributes, and thus can generate multi-view consistent images with modified attributes. However, we observe that editing of some attributes can affect other unrelated attributes. This is especially evident for the local attributes. For example, after converting the “No-Smile” face into “Smile”, the face identity can be modified. We believe this is because the latent codes for some attributes of face images are strongly entangled. To alleviate this problem, we propose a novel method, ‘Training-as-Init, Optimizing-for-Tuning’ (TRIOT), to search for better latent codes that provide better non-target region preservation and more meaningful target-region editing. We first employ the *off-the-shelf* face parsing model [Yu et al. 2018] to obtain the attribute-specific mask \mathbf{M} . Given the mask, we select the non-target region \mathbf{M} and the target region $1 - \mathbf{M}$. As shown in the left of Fig. 3, we can get: I_r , I_r^n , I_e , I_e^n , and the initial latent code \mathbf{w}_e . Given the corresponding optimization objection, we desire to find an optimal latent code $\hat{\mathbf{w}}_e$ and the corresponding image \hat{I}_e and normal \hat{I}_e^n .

Specifically, our objection functions are attribute-specific, consisting of two parts: geometry consistency loss and texture consistency loss. As shown in the right of Fig. 3, our texture and geometry consistency optimization objection is defined as:

$$\begin{aligned}\hat{\mathbf{w}}_e &= \arg \min_{\hat{\mathbf{w}}_e} \|\mathbf{M} \odot (\hat{I}_e - I_r)\|_1 + \|(1 - \mathbf{M}) \odot (\hat{I}_e - I_e)\|_1, \\ &+ \|\mathbf{M} \odot (\hat{I}_e^n - I_r^n)\|_1 + \|(1 - \mathbf{M}) \odot (\hat{I}_e^n - I_e^n)\|_1.\end{aligned}\quad (13)$$

This texture loss is used to reduce differences between the reconstructed texture I_r and the optimized texture \hat{I}_e in the non-target region while keeping \hat{I}_e the same as I_e in the target region for the attribute-editing. The geometry consistency loss has the same objective function as the texture loss.

We can obtain better editing results and preserve the non-target regions after less than 1,000 iterations by optimizing these consistency losses as the objective function. Note that we do not apply this method for all attributes, as the first step can achieve acceptable editing results, especially for global attributes. Thus, we can allow our method to balance the training speed and editing quality. Which is not the case for the training or optimization only methods. Moreover, the consistency loss is attribute-specific. We take the attribute “Smile” as an example in Fig. 3. In reality, the geometry loss is simpler than the objective function above for some attributes, such as “Hair Color”, as the optimized normal should be the same as the reconstructed normal in the hair color editing. Thus, we do



Fig. 4. Visual results of face attribute editing and the multi-view renderings from our TT-GNeRF (E) and TT-GNeRF (S). We use attributes “Hair Color”, “Gender”, “Age”, “Smile”, “Bangs” as the example (Zoom in for best view.). TT-GNeRF (S) and TT-GNeRF (E) mean our method with different backbones: StyleSDF and EG3D, respectively.

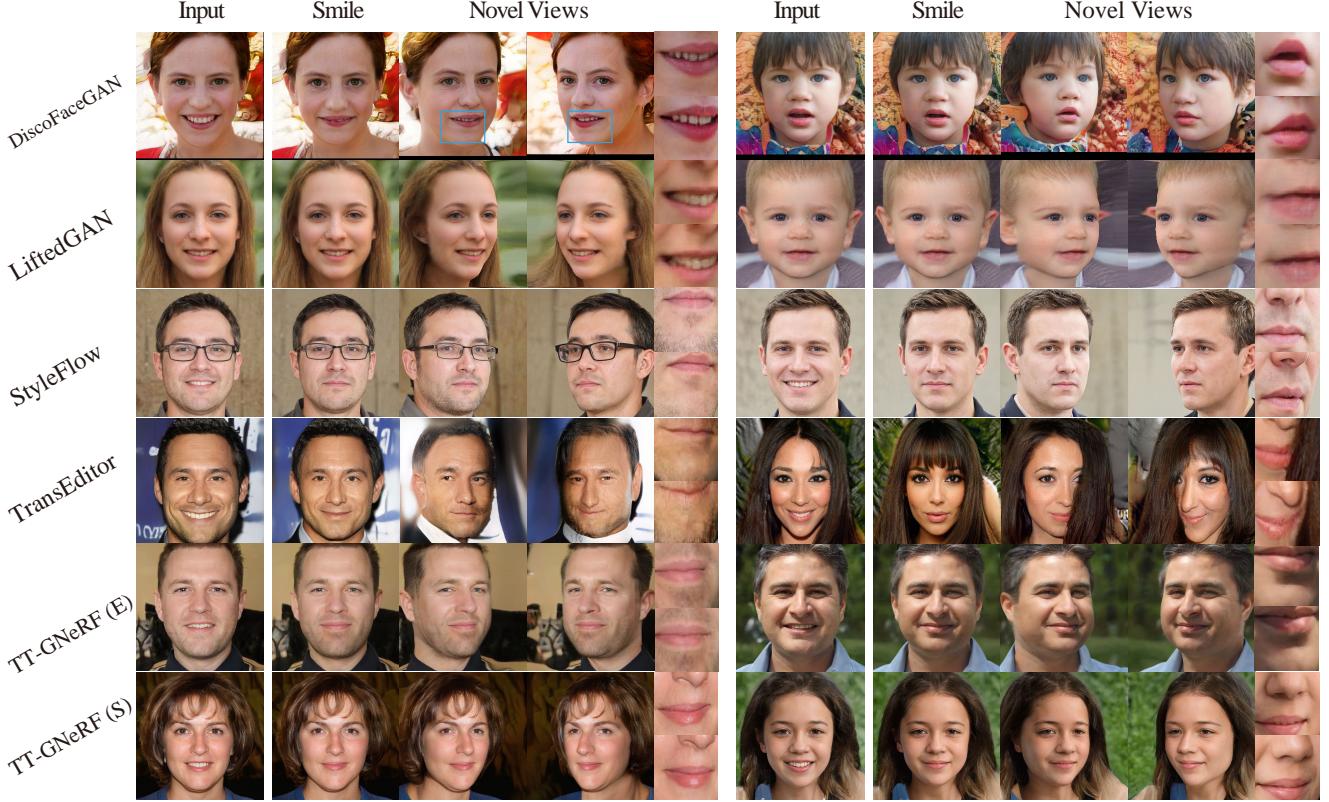


Fig. 5. Qualitative comparisons between our method and the baselines, *i.e.*, DiscoFaceGAN [Deng et al. 2020], LiftedGAN [Shi et al. 2021], StyleFlow [Abdal et al. 2021], and TransEditor [Xu et al. 2022b] on “Smile” attribute and the corresponding multiple-view renderings.

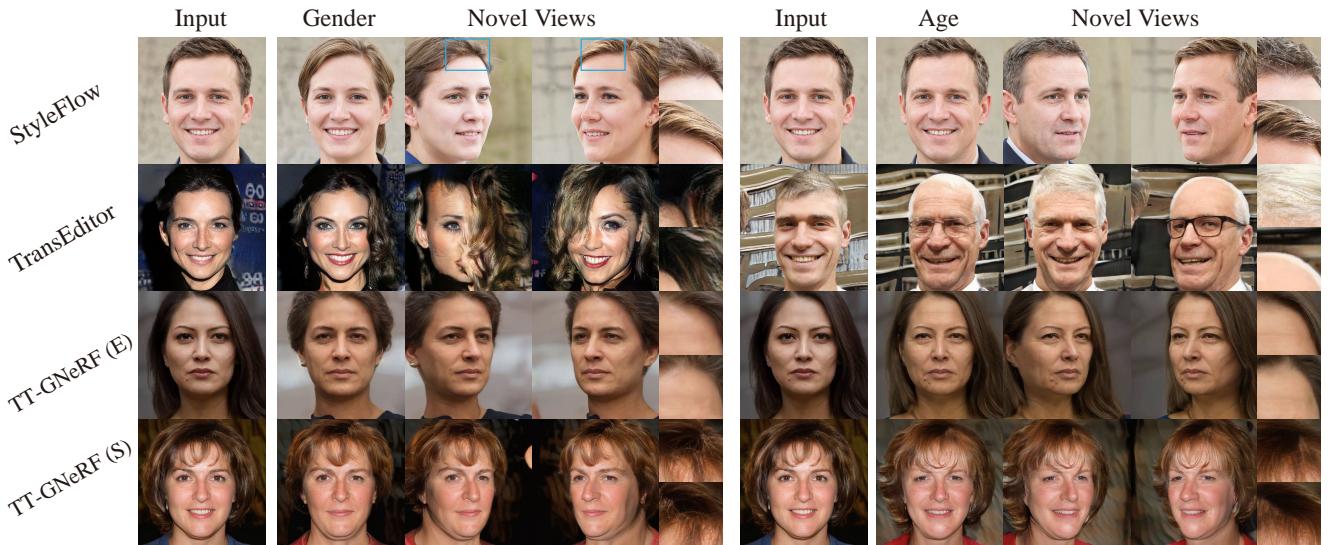


Fig. 6. Qualitative comparisons between our method and the baselines, *i.e.*, StyleFlow [Abdal et al. 2021], and TransEditor [Xu et al. 2022b] on “Gender” and “Age” attributes and the corresponding multiple-view renderings. Note that DiscoFaceGAN and LiftedGAN cannot deal with these two attributes.

not need to use the mask to define the foreground and background region for the “Hair Color” attribute.

3.4 Geometry Editing

Our model can edit geometry while preserving the appearance of the face. Similar to the TRIOT method, we utilize a similar optimization method to search the latent code with a different geometry but with

Table 1. Quantitative results on the attributes editing results using four metrics: FID, Classification Accuracy (CA), Average Matching Point (aMP), Face Recognition Similarity (FRS), and Local Preservation (LP).

Method	Smile					Gender					Age				
	FID ↓	CA ↑	aMP ↑	FRS ↑	LP ↓	FID ↓	CA ↑	aMP ↑	FRS ↑	FID ↓	CA ↑	aMP ↑	FRS ↑		
DiscoFaceGAN [Deng et al. 2020]	77.84	55.00	1347.4	0.587	7.280	-	-	-	-	-	-	-	-	-	-
LiftedGAN [Shi et al. 2021]	95.25	-	1484.0	0.464	-	-	-	-	-	-	-	-	-	-	-
StyleFlow [Abdal et al. 2021]	78.98	99.24	1089.7	0.586	19.36	82.80	79.90	1088.4	0.588	95.61	89.9	1090.8	0.586		
TransEditor [Xu et al. 2022b]	55.97	90.23	1075.6	0.564	40.23	56.60	76.73	964.30	0.575	78.32	99.5	959.82	0.490		
TT-GNeRF (E)	64.83	93.20	1527.5	0.852	18.20	65.84	86.00	1528.6	0.825	63.09	79.2	1554.5	0.864		
TT-GNeRF (S)	56.37	88.70	1899.6	0.812	5.870	55.74	73.40	1825.1	0.822	55.43	81.7	1982.8	0.850		

a similar texture. As shown in the right of Fig. 3, given the original latent code \mathbf{w} and another code $\tilde{\mathbf{w}}$, the corresponding face image and normal pairs are $(\mathbf{I}, \mathbf{I}^n)$ ($\tilde{\mathbf{I}}, \tilde{\mathbf{I}}^n$), respectively. We use $\tilde{\mathbf{w}}$ as the initial code of the optimization pipeline. The optimization objective is defined as:

$$\tilde{\mathbf{w}} = \arg \min_{\tilde{\mathbf{w}}} \lambda_3 LPIPS(\mathbf{I}, \tilde{\mathbf{I}}) + 1 - \|(\mathbf{I}^n - \tilde{\mathbf{I}}^n)\|_1. \quad (14)$$

This optimization stage can efficiently find a latent code with a different geometry but similar texture within hundreds of steps.

4 EXPERIMENTS

We name our method TT-GNeRF (training and tuning generative neural radiance field) and use TT-GNeRF (S) and TT-GNeRF (E) to refer to our method with two different backbones: StyleSDF [Or-El et al. 2022] and EG3D [Chan et al. 2022], respectively.

4.1 Setting

Dataset. Given that the two backbones (*i.e.*, StyleSDF and EG3D) are pretrained on the FFHQ dataset [Karras et al. 2019], we train the model using the sampled images and the corresponding latent codes. We use 100,000 image for training StyleSDF and 40,000 for EG3D. We use the *off-the-shelf* attribute-classifiers [Karras et al. 2020] to obtain several attribute labels, including Hair Color, Gender, Bangs, Age, Smile, and Beard. We use all generated triplets to train our models.

Implementation Details. In order to speed up the training and reduce memory consumption, we compute the reconstruction and adversarial losses using the low-resolution images, which are rendered from RGB values \mathbf{c} . We set $\lambda_1 = 10$, $\lambda_2 = 1.0$ for the classification and reconstruction losses. We train DAEM with Adam optimizer [Kingma and Ba 2014] $\beta_1=0.0$, $\beta_2=0.99$, and learning rate= $1e-4$ for 50,000 steps. For the TRIOT, we set $\lambda_3 = 0.3$ of Eq. 14. We also use Adam optimizer with $\beta_1=0.9$, $\beta_2=0.99$, and learning rate= $5e-4$. In this stage, we train for 1000 optimization steps. Our discriminator has an architecture similar to StyleGAN2 [Karras et al. 2020], but has an additional classification branch. To improve the stability of the adversarial training, we adopt the non-saturating logistic loss [Goodfellow et al. 2014] and R1 regularization [Mescheder et al. 2018].

Compared Baselines. Since our method is an attribute-conditioned generative model, the most similar supervised method is StyleFlow [Abdal et al. 2021], which can be used for face attribute editing tasks and multiple-view generation. Moreover, we adopt the state-of-the-art generative model, TransEditor [Xu et al. 2022b] as our baseline for comparing face semantic disentanglement with multiple-view generation results. We also compare with the 3DMM-guided model, DiscoFaceGAN [Deng et al. 2020]. Note that this model can only edit some expression-related attributes, such as “Smile”. Finally, we also adopt a 3D-aware LiftedGAN [Shi et al. 2021] to compare multiple-view generation. However, LiftedGAN cannot control individual attributes.

Evaluation Metrics. We use five metrics for evaluation: FID (Fréchet Inception Distance) score [Heusel et al. 2017] to evaluate the quality and diversity of the edited images. We use Classification Accuracy (CA) to evaluate the correctness of the edited attributes. Moreover, we use average Matching points (aMP) [Zhang et al. 2021] and Face Recognition Similarity (FRS) [Liu et al. 2021a] to evaluate the consistency of multiple-view generation results quantitatively. Additionally, we evaluate the non-target region preservation of the editing result by the Local Preservation (LP).

To evaluate the quality and diversity of the edited results, we calculate the FID score [Heusel et al. 2017] by using the samples in FFHQ as the real distribution, and using the original image and its edited results as the fake distribution. We sample 5000 real and fake samples from all models for each attribute to calculate FID scores. A lower FID score indicates a lower discrepancy between the image quality of the real and generated images.

To evaluate the accuracy of the attribute transfer, we use the *off-the-shelf* classifiers [Karras et al. 2020] to classify the edited samples and compute the accuracy by comparing the predicted and the target labels. We refer to this metric as Classification Accuracy (CA). We calculate the CA, 1000 edited samples from all models for each attribute. Higher is better for CA.

As is well known, it is a challenging task to evaluate the view consistency without ground truth. Thus, we rely on proxy metrics to evaluate it. We follow 3D-SGAN [Zhang et al. 2021] and use the average Matched Points metric *aMP*. In previous work, 3D-SGAN [Zhang et al. 2021] uses Patch2Pix [Zhou et al. 2021a] to compute a point-wise matching between two images (I_1, I_2) generated from the same identity with different viewpoints. Then, they count the number of Matched Points $MP(I_1, I_2)$. In our work, we

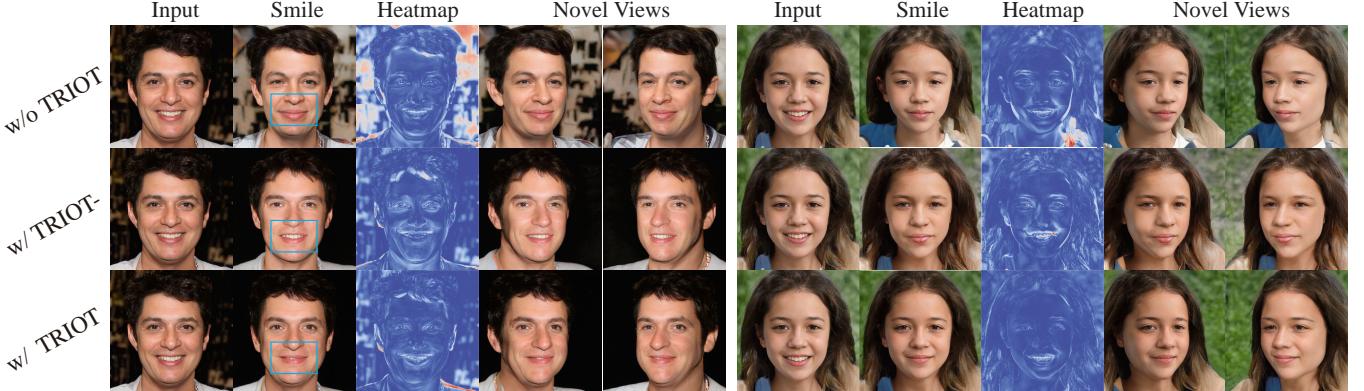


Fig. 7. Ablation study for our TRIOT method. The results are from TT-GNeRF (S) with “Smile” as the target attribute. We visualize the differences between input and edited images using a “coolworm” heatmap. TRIOT- means the proposed optimization objective excludes geometry consistency loss.

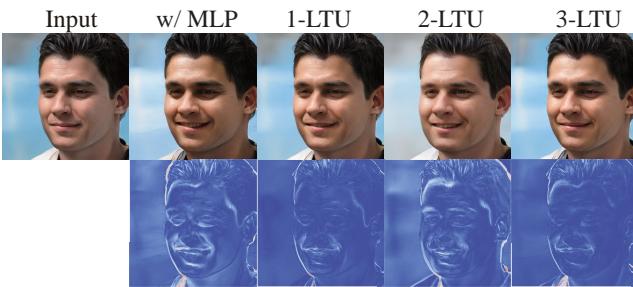


Fig. 8. Ablation study for Attribute editing module (AEM). We visualize the “Smile” editing results corresponding to the difference heatmaps between each edited image and the input image.

chose 100 random identities and generated ten different views for each identity. Finally, we calculate the mean of *MP* across all pairs of samples and get a final *average MP* (aMP) score. Additionally, we utilize Face Recognition Similarity (FRS) [Liu et al. 2021a] to evaluate the identity preservation in the different views. In detail, we use the state-of-the-art face recognition method ArcFace [Deng et al. 2019] to estimate the feature similarity of two facial images and compute the average score across 1000 samples with ten different views and 100 identities. The higher aMP and FRS scores indicate that the synthesized images with different viewpoints have more similar identities to the input faces.

We use 1000 input-edited paired samples to evaluate the Local Preservation score (LP) for the local attribute, such as “Smile”. For every paired sample, we use the *off-the-shelf* face parsing network [Yu et al. 2018] to collect the corresponding mask. Then, we use ℓ_1 distance to measure the differences of each paired sample and average them across all pairs.

We notice that LiftedGAN and DiscoFaceGAN cannot directly perform editing tasks, such as “Gender” and “Age”. Thus, we do not provide scores for these methods. Besides, we provide the FID, aMP, and FRS scores of LiftedGAN by using their randomly generated samples instead of the attribute-edited samples.

4.2 State-of-the-Art Comparison

Fig. 4 shows that our method can achieve high-quality face editing results with accurate attribute transfer, and non-target region

Table 2. Ablation study for our TRIOT.

Method	CA \uparrow	aMP \uparrow	FRS \uparrow	LP \downarrow
w/o TRIOT	85.0	1645.1	0.869	11.3
w/ TRIOT-	83.4	1766.9	0.832	10.8
w/ TRIOT	82.0	1706.3	0.854	7.56

Table 3. Ablation study for Attribute editing module (AEM). We use the attribute “Smile” for this comparison.

Method	FID \downarrow	CA \uparrow	LP \downarrow
w/ MLP	57.99	86.80	6.62
1-LTU	56.37	88.70	5.87
2-LTU	58.63	93.20	6.87
3-LTU	57.71	88.70	13.33

preservation. For example, the first row of Fig. 4 shows the hair-color transfer from brown hair to black hair. We can observe that the face identity, including expression, is not changed overall. Moreover, the 3-8 columns show the multiple-view generation of edited results, demonstrating that our results show strong 3D consistency. Comparing the results of TT-GNeRF (S) to TT-GNeRF (E), the later tends to learn a wider variety in appearance for global attributes, such as “Gender”.

We compare our method with the baselines on facial attribute editing with multiple-view generation in Fig. 5 and Fig. 6. As aforementioned, we take “Smile”, “Gender” and “Age” attributes as examples, as most baselines can directly edit these three attributes. Fig. 5 shows the “Smile” editing results. We can observe that most models can achieve accurate transfer from “Smile” to “No-Smile” while preserving the non-target region well. However, TransEditor [Xu et al. 2022b] fails to preserve the non-target region, as shown in the 4-th rows of Fig. 5. Moreover, our method outperforms all baselines in the 3D consistency for novel-view generation of the edited results. Specifically, DiscoFaceGAN cannot keep the hair color when changing the pose, and the zoomed-in mouth shows the expression has been changed compared to the original view. StyleFlow and TransEditor fail to handle large pose variations and suffer from the severe view-inconsistency problem. The identity has changed a lot for both models, such as the beard of the zoomed-in mouth.

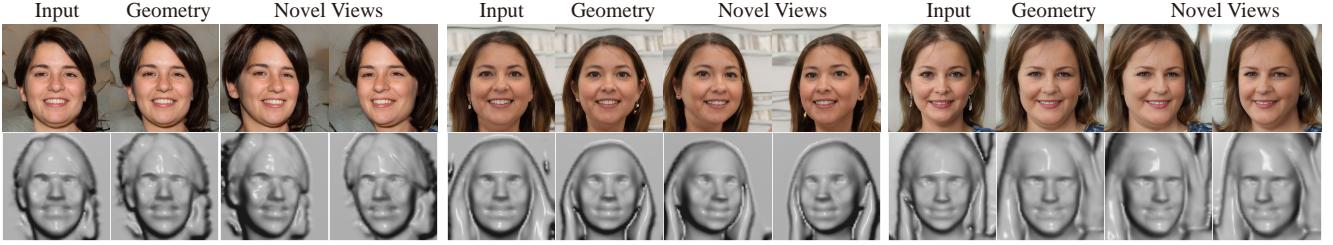


Fig. 9. Visual results of Geometry editing from TT-GNeRF (S).

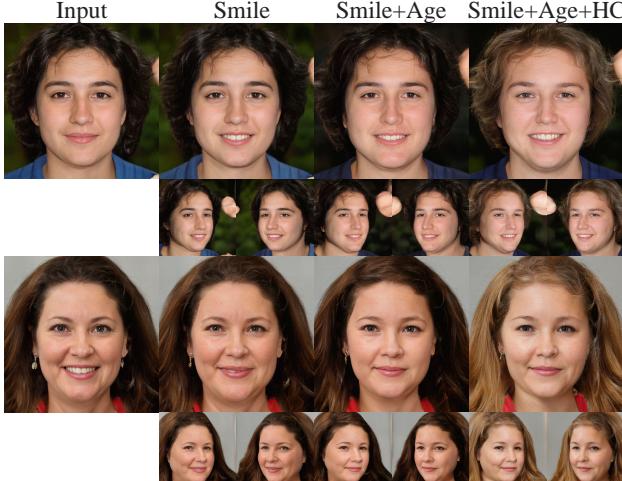


Fig. 10. Multiple-attribute editing results from our TT-GNeRF (S) method. HC: Hair Color.

On the other hand, LiftedGAN has improved 3D consistency but has the limited quality and cannot perform facial attribute editing. Fig. 6 shows “Gender” and “Age” editing results that demonstrate the superiority of our method in the 3D consistency compared to other methods. Please check the zoomed-in hair region for detailed comparisons.

Table 1 shows the quantitative evaluation results of “Smile”, “Gender”, and “Age” attributes editing results. For the FID score, our models achieve comparable performance with all three attributes’ baselines. Specifically, our TT-GNeRF (S) obtains the best FID scores for both the “Gender” and “Age” attributes and achieves a comparable score with TransEditor for the “Smile” attribute. Moreover, our models are competitive with the baselines on CA. For example, TT-GNeRF (E) achieves 86.00, compared to 79.90 of StyleFlow, and 76.73 of TransEditor for the “Gender” attribute. However, for the “Age” attribute, both models are worse than the previous baselines. We guess this is because the 3D-Aware GAN has worse disentanglement than the 2D method, especially for the “Age” attribute. Table 1 shows that our models outperform previous methods on both aMP and FRS metrics for the three attributes. Specifically, for the “Smile” attribute, our TT-GNeRF (S) achieve 1899.6 aMP and 0.812 FRS scores, which are better than 1484.0 aMP and 0.464 FRS scores of LiftedGAN, 1347.4 aMP, and 0.587 FRS scores of DiscoFaceGAN.

Overall, our qualitative and quantitative results demonstrate the effectiveness of face attribute editing and the superior 3D consistency of the multiple-view for editing results.

4.3 Ablation Study

Latent Transfer Unit (LTU). In our attribute editing module (AEM), we utilize LTU for more accurate editing of the target region, alternatively one can adopt MLP for this task. Thus, to better understand the effect of LTU on the final performance, we show four variants of the model: w/ MLP, 1-LTU, 2-LTU, 3-LTU; w/ MLP means the model without LTU, but with two-layers MLP instead; 1-LTU, 2-LTU, 3-LTU are the models with one, two or three LTU layers respectively. Fig. 8 shows the comparison for all variants. We can observe that LTU variants edit the expression of the input while better preserving the non-target region than the variant w/ MLP. Table 3 shows quantitative scores from these four variants. We can observe that the variant with 1-LTU can achieve the best scores in terms of FID and LP metrics, and it is better than w/ MLP in the CA metric. Moreover, multiple-layer LTU can harm the model performance, especially the preservation ability of the non-target regions. Thus, we select the variant with 1-LTU for all other experiments.

Training-As-Init, Optimizing for Tuning (TRIOT). Our proposed TRIOT can further improve the identity preservation after the face attribute editing, especially for local attributes. To validate it, we compare the TRIOT with two ablation baselines: w/o TRIOT, and w/ TRIOT-. The baseline w/o TRIOT eliminates the optimization stage, and w/ TRIOT- means the optimization objective excludes the geometry consistency loss. Fig. 7 provides the comparisons that showcase the effectiveness of our TRIOT. We can observe that our model can change the face attribute from “Smile” to “No-Smile”, and the variant with TRIOT shows better preservation of the non-target region than the variant w/o TRIOT. It can be clearly seen from the styles of bangs and the corresponding heatmaps. Compared to w/ TRIOT-, w/ TRIOT achieves a better balance between the editing and preservation, and w/ TRIOT- suffers from inaccurate editing in the geometry (Left of Fig. 7). To quantitatively demonstrate the effectiveness of the TRIOT, we sample 100 test samples to compute the metrics, which are presented in Table 2. Here, the variant with the TRIOT shows the best LP score, which is consistent with the visual results of Fig. 7. Moreover, three variants achieve similar scores in other metrics, i.e., CA, aMP, and FRS, which demonstrates that our TRIOT does not hurt the editing ability and the view consistency of the model.

4.4 Applications

Multiple-attribute Editing. In addition to the previous single-attribute edits, shown in Fig. 4, our model also can achieve sequential editing of multiple attributes. Fig. 10 shows our high-quality edits for the sequence “Smile + Age + Hair color”. At the bottom of each

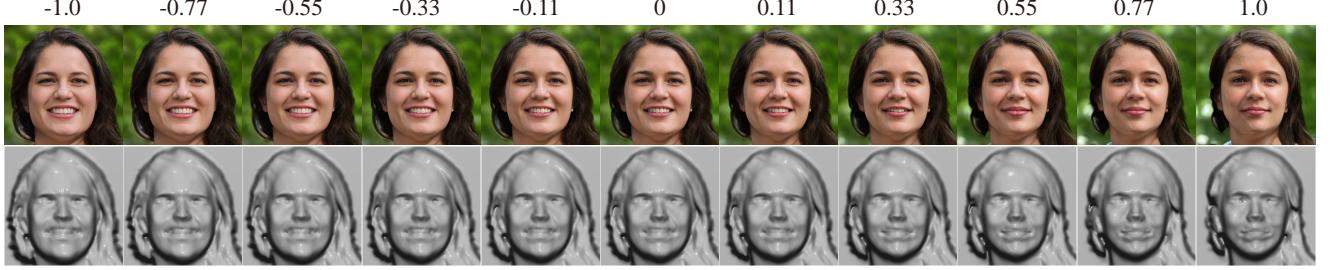


Fig. 11. Label interpolation from -1 to 1 for face attribute editing. The results are from TT-GNeRF (S) with “Smile” as the target attribute.

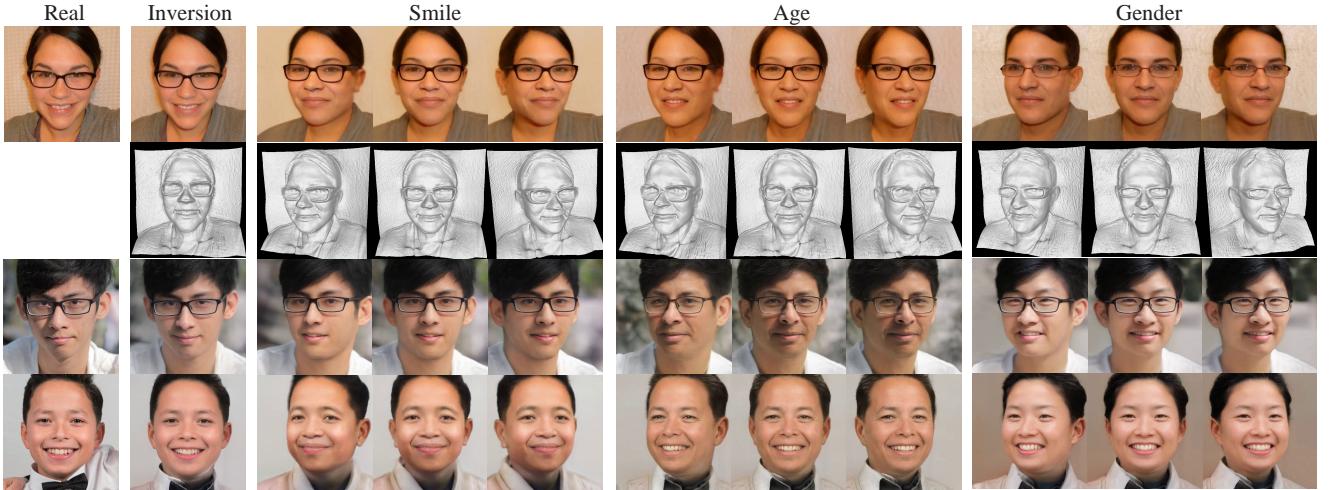


Fig. 12. GAN Inversion for real image editing and corresponding multiple-view generation. The results are from TT-GNeRF (E) with “Smile”, “Gender” and “Age” as the target attributes.

row, the multiple-view results for these edits are presented, which shows that our model still shows strong 3D-view consistency in these cases.

Geometry Editing. The disentanglement of the geometry and textures is not easy to achieve in previous methods. As mentioned above, we propose a simple unsupervised optimization method for geometry editing. Fig. 9 shows our geometry editing results, corresponding multiple-view results, and meshes, respectively. For the middle case, we can observe the face has been reduced in size, while the appearance (e.g., hair color and expression) are well preserved.

Label Interpolation. We show the attribute transfer results by continuously interpolating the attribute label. We take the attribute “Smile” as an example. Fig. 11 shows the interpolation results corresponding to the labels ranging from -1 to 1. We can observe that the facial expression has been changed gradually from “Smile” to “No-Smile”, while the identity (including the geometry) has minor changes. As mentioned above, the proposed TRIOT method can be used to alleviate this problem.

GAN Inversion for Real Image Editing. We utilize the state-of-the-art GAN Inversion method (PTI) to project real images into the latent space of our 3D-GAN. Then, we perform real image editing with our proposed DAEM and TRIOT methods. We show the results of the PTI method for TT-GeNRF (E). PTI for TT-GNeRF (S) suffers from low-quality generation and editing. We observe that tuning

step of PTI can harm the geometry of StyleSDF. In detail for TT-GNeRF (E), we follow two steps of PTI: 1) optimizing the latent code to obtain the corresponding projected latent of the real image while fixing the generator parameters (including DAEM); 2) fixing the optimized latent code and the parameters of DAEM while finetuning the remaining parameters of the generator. Afterward, we perform image editing using DAEM for the projected latent code.

Fig. 12 shows real image inversion and editing results for “Gender”, “Age” and “Smile” attributes. The 2-th column shows that we can produce almost perfect reconstruction for the real images. After that, our model can achieve realistic and accurate editing. Moreover, the non-target region is well preserved. Finally, our model’s multiple-view results also show strong 3D consistency.

We refer to the demo video for more results about multiple-view attribute editing, geometry editing, and GAN inversion for real image editing.

5 CONCLUSIONS

In this work, we propose an attribute-conditional 3D-aware face generating and editing model, which shows the disentangling abilities of the generative neural radiance field with attributes as inputs. Moreover, we integrate the training method for the proposed DAEM and the optimization method (TRIOT) into the 3D-aware face editing task to balance the best trade-off between quality and efficiency. Our

model can achieve higher-quality 3D-aware face attribute editing compared to previous methods while better preserving the 3D consistency for different view generations. The qualitative and quantitative results demonstrate the superiority of our method. Additionally, our model achieves geometry editing with the simple optimization method while preserving the appearance.

However, there still exist some limitations. First, our model fails in editing the facial attribute in some cases. For example, the age column of Table 1 shows that our CA score is worse than some methods. Future work could alleviate this by learning a better classifier in the discriminator. Second, our proposed TRIOT still costs some minutes for single attribute editing; thus, it is unacceptable for some real application scenarios. Finally, as shown in Fig. 12, our model can achieve the single image 3D model and perform attribute editing. However, compared to video-based head avatars [Gafni et al. 2021; Zheng et al. 2022], the identity is not well preserved between real images and projected images. Proposing better GAN inversion techniques adapted for 3D-Aware GAN can further alleviate this problem.

REFERENCES

- Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2stylegan: How to embed images into the stylegan latent space?. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4432–4441.
- Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. Image2stylegan++: How to edit the embedded images?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8296–8305.
- Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. 2021. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (TOG)* 40, 3 (2021), 1–21.
- Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6711–6720.
- Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5855–5864.
- Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *CVPR* (2022).
- Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. 2021. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *CVPR*.
- Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. 2022. Efficient geometry-aware 3D generative adversarial networks. *CVPR* (2022).
- Xuanhong Chen, Bingbing Ni, Naiyuan Liu, Ziang Liu, Yiliu Jiang, Loc Truong, and Qi Tian. 2020. Coogan: A memory-efficient framework for high-resolution facial attribute editing. In *European Conference on Computer Vision*. Springer, 670–686.
- Yuedong Chen, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. 2022. Sem2NeRF: Converting Single-View Semantic Masks to Neural Radiance Fields. *ECCV* (2022).
- Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*.
- Wenqing Chu, Ying Tai, Chengjin Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. 2020. SSCGAN: Facial Attribute Editing via Style Skip Connections. (2020).
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
- Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. 2020. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5771–5780.
- Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4690–4699.
- Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. 2020. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5154–5163.
- Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. 2022. GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation.
- Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8649–8658.
- Yue Gao, Fangyun Wei, Jianmin Bao, Shuyang Gu, Dong Chen, Fang Wen, and Zhouhui Lian. 2021. High-fidelity and arbitrary face editing. In *CVPR*.
- Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 2019. 3d guided fine-grained face manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9821–9830.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
- Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. 2022. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *ICLR* 2022 (2022).
- Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems* 33 (2020), 9841–9850.
- Zhenliang He, Meina Kan, and Shiguang Shan. 2021. Eigengan: Layer-wise eigen-learning for gans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14408–14417.
- Zhenliang He, Meina Kan, Jichao Zhang, and Shiguang Shan. 2020. PA-GAN: Progressive Attention Generative Adversarial Network for Facial Attribute Editing. *arXiv preprint arXiv:2007.05892* (2020).
- Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. Attnan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing* 28, 11 (2019), 5464–5478.
- Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. 2019. Escaping plato’s cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9984–9993.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*.
- Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. 2022. Headnerf: A real-time nerf-based parametric head model. *CVPR* (2022).
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR* (2017).
- Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* 34 (2021).
- Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8110–8119.
- Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. 2021. Exploiting spatial dimensions of latent in gan for real-time image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 852–861.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *ICLR* (2014).
- Gihyun Kwon and Jong Chul Ye. 2021. Diagonal attention and style-based GAN for content-style disentanglement in image generation and translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 13980–13989.
- Hanbang Liang, Xianxu Hou, and Linlin Shen. 2021. SSFlow: Style-guided Neural Spline Flows for Face Image Manipulation. In *ACMMM*.
- Connor Z Lin, David B Lindell, Eric R Chan, and Gordon Wetzstein. 2022. 3D GAN Inversion for Controllable Portrait Image Animation. *arXiv preprint arXiv:2203.13441* (2022).
- Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. 2019. STGAN: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*. 3673–3682.
- Yahui Liu, Yajing Chen, Linchao Bao, Nicu Sebe, Bruno Lepri, and Marco De Nadai. 2021a. ISF-GAN: An Implicit Style Function for High-Resolution Image-to-Image Translation. *TMM* (2021).
- Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. 2021b. Smoothing the Disentangled Latent Style Space for Unsupervised Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics* 21, 4 (1987), 163–169.
- Li Ma, Xiaoyu Li, Jing Liao, Xuan Wang, Qi Zhang, Jue Wang, and Pedro Sander. 2022. Neural Parameterization for Dynamic Human Head Editing. *TOG* (2022).
- Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which training methods for GANs do actually converge?. In *International conference on machine learning*.

- PMLR, 3481–3490.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.
- Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *TOG* (Jan. 2022).
- Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7588–7597.
- Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. 2022. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Michael Niemeyer and Andreas Geiger. 2021. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. In *CVPR*.
- Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. 2022. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. *CVPR2022* (2022).
- Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. 2021. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. *NeurIPS* (2021).
- Pascal Payan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*. Ieee, 296–301.
- William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. 2020. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*. Springer, 581–597.
- Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021a. Animatable Neural Radiance Fields for Human Body Modeling. In *ICCV*.
- Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *CVPR*.
- Christian Reiser, Songyu Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *ICCV*.
- Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2287–2296.
- Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2021. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744* (2021).
- Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. 2020. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *NeurIPS*.
- Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. 2022. VoxGRAF: Fast 3D-Aware Image Synthesis with Sparse Voxel Grids. *ARXIV* (2022).
- Yujun Shen and Bolei Zhou. 2021. Closed-form factorization of latent semantics in gans. In *CVPR*. 1532–1540.
- Yichun Shi, Divyansh Aggarwal, and Anil K Jain. 2021. Lifting 2D StyleGAN for 3D-Aware Face Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6258–6266.
- Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. 2022. SemanticStyleGAN: Learning Compositional Generative Priors for Controllable Image Synthesis and Editing. *CVPR* (2022).
- Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. 2022. EpiGRAF: Rethinking training of 3D GANs. *arXiv preprint arXiv:2206.10535* (2022).
- Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. 2022a. IDE-3D: Interactive Disentangled Editing for High-Resolution 3D-aware Portrait Synthesis. *TOG* (2022).
- Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. 2022b. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7672–7682.
- Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. 2022. Block-NeRF: Scalable Large Scene Neural View Synthesis. *arXiv preprint arXiv:2202.05263* (2022).
- Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. 2020. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6142–6151.
- Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.
- Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. 2022. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. *CVPR* (2022).
- Andrey Voynov and Artem Babenko. 2020. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*. PMLR, 9786–9796.
- Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. 2021. Orthogonal Jacobian Regularization for Unsupervised Disentanglement in Image Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6721–6730.
- Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. 2019. Relgan: Multi-domain image-to-image translation via relative attributes. In *CVPR*.
- Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. 2022. GRAM-HD: 3D-Consistent Image Generation at High Resolution with Generative Radiance Manifolds. *arXiv preprint arXiv:2206.07255* (2022).
- Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 2022a. 3D-aware Image Synthesis via Learning Structural and Textural Representations. *CVPR* (2022).
- Yanbo Xu, Yueqin Yin, Liming Jiang, Qianyi Wu, Chengyao Zheng, Chen Change Loy, Bo Dai, and Wayne Wu. 2022b. TransEditor: Transformer-Based Dual-Space GAN for Highly Controllable Facial Editing. *CVPR* (2022).
- Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. 2022. GIRAFFE HD: A High-Resolution 3D-aware Generative Model. *CVPR* (2022).
- Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. 2021. iNeRF: Inverting Neural Radiance Fields for Pose Estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 325–341.
- Jichao Zhang, Enver Sangineto, Hao Tang, Aliaksandr Siarohin, Zhun Zhong, Nicu Sebe, and Wei Wang. 2021. 3D-Aware Semantic-Guided Generative Model for Human Synthesis. *arXiv preprint arXiv:2112.01422* (2021).
- Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492* (2020).
- Xuanmeng Zhang, Zhedong Zheng, Dainhang Gao, Bang Zhang, Pan Pan, and Yi Yang. 2022. Multi-View Consistent Generative Adversarial Networks for 3D-aware Image Synthesis. *CVPR* (2022).
- Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühl, Xu Chen, Michael J Black, and Otnar Hilliges. 2022. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13545–13555.
- Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. 2021b. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. (2021). *arXiv:2110.09788*
- Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. 2021a. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*.
- Jiapeng Zhu, Yujun Shen, Yinghao Xu, Deli Zhao, and Qifeng Chen. 2022. Region-Based Semantic Factorization in GANs. *CVPR* (2022).
- Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. 2020. In-domain gan inversion for real image editing. In *European conference on computer vision*. Springer, 592–608.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*.
- Yiyu Zhuang, Hao Zhu, Xusen Sun, and Xun Cao. 2021. MoFaNeRF: Morphable Facial Neural Radiance Field. *arXiv preprint arXiv:2112.02308* (2021).