# PREIM3D: 3D Consistent Precise Image Attribute Editing from a Single Image

Jianhui Li[1,2], Jianmin Li[1] * , Haoji Zhang[1], Shilong Liu[1], Zhengyi Wang[1],
Zihao Xiao[3], Kaiwen Zheng[1], Jun Zhu[1] *.

[1] Department of Computer Science and Technology, Institute for AI, BNRist, Tsinghua University
[2] State Key Laboratory of Astronautic dynamics, Xi'an Satellite Control Center    [3] RealAI

Figure 1. 3D consistent precise inversion and editing. Our method enables reconstructing texture and geometry from a single real image and allows one to perform a list of attributes editing sequentially. The yaw angles of the second to sixth columns are $[-30°, -20°, 0°, 20°, 30°]$. The last column is the shape of the sixth column.

| input | inversion | smile | black hair | age | eyeglasses | shape |

## Abstract

*We study the 3D-aware image attribute editing problem in this paper, which has wide applications in practice. Recent methods solved the problem by training a shared encoder to map images into a 3D generator's latent space or by per-image latent code optimization and then edited images in the latent space. Despite their promising results near the input view, they still suffer from the 3D inconsistency of produced images at large camera poses and imprecise image attribute editing, like affecting unspecified attributes during editing. For more efficient image inversion, we train a shared encoder for all images. To alleviate 3D inconsistency at large camera poses, we propose two novel methods, an alternating training scheme and a multi-view identity loss, to maintain 3D consistency and subject identity. As for imprecise image editing, we attribute the problem to the gap between the latent space of real images and that of generated images. We compare the latent space and inversion manifold of GAN models and demonstrate that editing in the inversion manifold can achieve better results in both quantitative and qualitative*

*evaluations. Extensive experiments show that our method produces more 3D consistent images and achieves more precise image editing than previous work. Source code and pretrained models can be found on our project page: https://mybabyyh.github.io/Preim3D/.*

## 1. Introduction

Benefiting from the well-disentangled latent space of Generative Adversarial Networks (GANs) [12], many works study GAN inversion [1, 2, 11, 28, 35, 36, 40] as well as real image editing in the latent space [14, 15, 22, 31, 32]. With the popularity of Neural Radiance Fields (NeRF) [24], some works start to incorporate it into GAN frameworks for unconditional 3D-aware image generation [6, 7, 13, 25–27, 30]. In particular, EG3D [6], the state-of-the-art 3D GAN, is able to generate high-resolution multi-view-consistent images and high-quality geometry conditioned on gaussian noise and camera pose. Similar to 2D GANs, 3D GANs also have a well semantically disentangled latent space [6, 13, 21, 33], which enables realistic yet challenging 3D-aware image editing.

Achieving 3D-aware image editing is much more chal-

---
* Corresponding authors.

lenging because it not only has to be consistent with the input image at the input camera pose but also needs to produce 3D consistent novel views. Recently, 3D-Inv [21] uses pivotal tuning inversion (PTI) [29], first finding out a pivotal latent code and then finetuning the generator with the fixed pivotal latent code, to obtain the latent code and edit the image attributes in the latent space. IDE-3D [33] proposes a hybrid 3D GAN inversion approach combining texture and semantic encoders and PTI technique, accelerating the optimization process by the encoded initial latent code. Pixel2NeRF [5] is the first to achieve 3D inversion by training an encoder mapping a real image to the latent space $\mathcal{Z}$ of $\pi$-GAN [7]. However, these methods still do not solve the problem of 3D consistency at large camera poses and precise image attribute editing. As shown in Fig. 4, some inverted images meet head distortion at large camera poses, or some unspecific attributes of edited images are modified.

In this paper, we propose a pipeline that enables **PR**ecise **E**diting in the **I**nversion **M**anifold with **3D** consistency efficiently, termed **PREIM3D**. There are three goals to achieve for our framework, (i) image editing *efficiently*, (ii) *precise inversion*, which aims to maintain realism and 3D consistency of multiple views, and (iii) *precise editing*, which is to edit the desired attribute while keeping the other attributes unchanged. 3D-Inv and IDE-3D optimized a latent code for each image, which is not suitable for interactive applications. Following Pixel2NeRF, we train a shared encoder for all images for efficiency.

To address *precise inversion*, we introduce a 3D consistent encoder to map a real image into the latent space $\mathcal{W}^+$ of EG3D, and it can infer a latent code with a single forward pass. We first design a training scheme with alternating in-domain images (i.e., the generated images) and out-domain images (i.e., the real images) to help the encoder maintain the 3D consistency of the generator. We optimize the encoder to reconstruct the input images in the out-domain image round. In the in-domain image round, we additionally optimize the encoder to reconstruct the ground latent code, which will encourage the distribution of the inverted latent code closer to the distribution of the original latent code of the generator. Second, to preserve the subject's identity, we propose a multi-view identity loss calculated between the input image and novel views randomly sampled in the surrounding of the input camera pose.

Though many works tried to improve the editing precision by modifying latent codes in $\mathcal{Z}$ space [31], $\mathcal{W}$ space [14, 17, 34], $\mathcal{W}^+$ space [1, 1, 40], and $\mathcal{S}$ space [37], they all still suffer from a gap between real image editing and generated image editing because of using the editing directions found in the original generative latent space to edit the real images. To bridge this gap, we propose a real image editing subspace, which we refer to *inversion manifold*. We compare the inversion manifold and the original latent

space and find the distortion between the attribute editing directions. We show that the editing direction found in the inversion manifold can control the attributes of the real images more precisely. To our knowledge, we are the first to perform latent code manipulation in the inversion manifold. Our methodology is orthogonal to some existing editing methods and can improve the performance of manipulation in qualitative and quantitative results when integrated with them. Figure 1 shows the inversion and editing results produced by our method. Given a single real image, we achieve 3D reconstruction and precise multi-view attribute editing.

The contributions of our work can be summarized as follows:

- We present an efficient image attribute editing method by training an image-shared encoder for 3D-aware generated models in this paper. To keep 3D consistency at large camera poses, we propose two novel methods, an alternating training scheme and a multi-view identity loss, to maintain 3D consistency and subject identity.

- We compare the latent space and inversion manifold of GAN models, and demonstrate that editing in the inversion manifold can achieve better results in both quantitative and qualitative evaluations. The proposed editing space helps to close the gap between real image editing and generated image editing.

- We conduct extensive experiments, including both quantitative and qualitative, on several datasets to show the effectiveness of our methods.

## 2. Related Work

### 2.1. NeRF-based GANs

NeRF models the underlying 3D scene through a continuous 5D function $F_\Theta$ that maps point $(x, y, z)$ and viewing direction $(\theta, \phi)$ to color and corresponding density, and then uses volume rendering techniques to render multiview images [24]. Although the standard NeRF requires multi-view images and trains the network for every single scene, several works [6, 7, 13, 25, 26, 30] combine it with the GAN framework to generate multi-view images from unconditional random samples. Among these, Holo-GAN [25] is the first NeRF-based GAN to learn 3D representations from unposed 2D images in an unsupervised manner. GRAF [30] produces high-resolution multi-view images of novel objects from disentangled shape code $z_s$ and appearance code $z_a$. GIRAFFE [26] can handle multi-object 3D scenes and control the synthesis of all objects separately. StyleNeRF [13] and EG3D [6] both utilize 2D CNN upsampling after neural rendering feature to achieve high-resolution 3D-aware images. Especially, EG3D [6] uses the

triplane representation to perform volume rendering, which is much more efficient than fully 3D networks. Our work is based on EG3D, which has a semantically disentangled latent space comparable to the state-of-the-art 2D generator, StyleGAN.

## 2.2. NeRF-based GAN Inversion

With the great success of GAN in the area of image generation, GAN inversion has become a popular research topic in image editing tasks. Existing GAN inversion methods can be grouped into three categories: optimization-based, encoder-based, and hybrid both. Optimization methods [1,2,9,20] directly optimize the latent code to minimize the distance between the generated image and the given image. Instead of optimization for every image, the encoder methods [4,15,28,35] train a category-specific generic encoder to map the given image to the latent code. pSp [28] extracts features from different pyramid scales encoding the image to style codes in $\mathcal{W}^+$ space. Further, e4e [35] analyzes the trade-off between distortion and editability. Based on e4e, HFGI [36] injects distortion residual features into the generator to improve the fidelity. Hybrid approach is a combination of encoder and optimization, leveraging the encoder to produce the initial latent code for optimization. The optimization-based and hybrid methods tend to favor faithful reconstruction over editability [28,36].

Despite the success of GAN inversion in 2D space, it is at its early age in 3D space. 3D GAN inversion is required to "imagine" the 3D geometry given only one single image, which is much more difficult than 2D GAN inversion. Recently, 3D-Inv [21] adopts PTI [29], a two-stage method, to perform the inversion. In the first stage, they optimize for the latent code $w$ reconstructing the face region by freezing the generator. In the second stage, they freeze the optimized latent code $w^*$, pivotal latent code, and fine-tune the generator by minimizing the LPIPS [39] loss. Further,IDE-3D [33] combines encoder and PTI, leveraging texture and semantic encoders to produce the pivotal latent code for acceleration. Since the PTI-based method requires optimization for each given image, it cost a long inference time and may converge to an arbitrary point in the latent space, which is not conducive to editing. Moreover, optimization on a single image leads to artifacts in novel views. In contrast, our method is a pure encoder requiring very little inference time, maintaining the perfect 3D consistency of the pretrained generator by fixing it.

Pixel2NeRF [5] is the first attempt to train a pure encoder mapping a single real image to the $\mathcal{Z}$ latent space in 3D GAN inversion task, which became a strong baseline. However, Pixel2NeRF suffers from attribute entanglement, low resolution, and 3D inconsistency. Different from Pixel2Nerf, we map the input image to $\mathcal{W}^+$ space, which is a more disentangled and editable space [20,37], and im-

prove the 3D consistency by explicitly encouraging multiple views. Table 1 outlines the differences between our method and previous methods.

## 2.3. Latent Space Manipulation

Numerous works have explored the GAN latent space for performing semantically manipulations in supervised or unsupervised manners [3, 8, 14, 17, 31, 34, 38]. The common approach is to seek the editing direction responsible for controlling a given attribute. GANSpace [14] adopts principal component analysis (PCA) to find the semantic direction. Sefa [32] directly decomposes the weight of the pretrained mapping network to discover the latent semantics with a closed-form factorization algorithm. InterfaceGAN [31] uses the normal vector of the boundary hyperplane of the SVM as the editing direction of the binary attribute. StyleSpace [37] proposes to use style channels to control a specific target attribute in $\mathcal{S}$ space which is defined by the channel-wise style parameters. To further disentangled the attribute editing, StyleFlow [3] trains a flow network formulating attribute-conditioned exploration as an instance of conditional continuous normalizing flows. These methods edit images in the original latent space of the generator, but we find that editing images in the inversion manifold achieves better qualitative and quantitative results.

## 3. Method

### 3.1. Overview

Current methods of 3D GAN inversion include encoder-based, optimization-based, and hybrid both. The optimization and hybrid methods are time-consuming, so they are not suitable for interactive applications such as avatar-based communication. Existing encoder-based methods often suffer from low resolution and inferior editability. To overcome these drawbacks, we introduce an encoder mapping a single real image to the latent space $\mathcal{W}^+$ and suggest performing image attribute editing in the inversion manifold.

Our method is based on a pretrained 3D generator such as EG3D, which can synthesize multi-view images conditioned on camera parameters $c$ and noise code $z \in \mathcal{Z} \subseteq \mathbb{R}^{512}$, where $z \sim \mathcal{N}(0,1)$. The mapping network of the generator transfers $z$ to an intermediate latent code $w \in \mathcal{W} \subseteq \mathbb{R}^{512}$, where $w$ is a 512-dimensional vector from a distribution without explicit formula. The generator $G$ takes in camera parameters $c$ and the latent code $w$ replicated $k$ times to synthesize images with the desired resolution as described:

$$X = G(w, c), \tag{1}$$

where $c$ denotes the camera parameters, including intrinsics and extrinsics. These replicated latent codes form the space $\mathcal{W}^+ \subseteq \mathbb{R}^{k*512}$. It has been shown that $k$ different style codes, rather than all the same style code, can increase

| | Inversion Type | Efficiently | 3D Consistency at Large Camera Poses | Superior Editability |
|---|---|---|---|---|
| IDE-3D [33] | hybrid | ✗ | ✗ | ✗ |
| 3D-Inv [21] | optimization-based | ✗ | ✗ | ✗ |
| Pixel2NeRF [5] | encoder-based | ✓ | ✗ | ✗ |
| PREIM3D (Ours) | encoder-based | ✓ | ✓ | ✓ |

Table 1. An overview of 3D GAN inversion and editing methods. 3D consistency and editability are evaluated by ID, APD, AA, and AD metrics described in section 4.
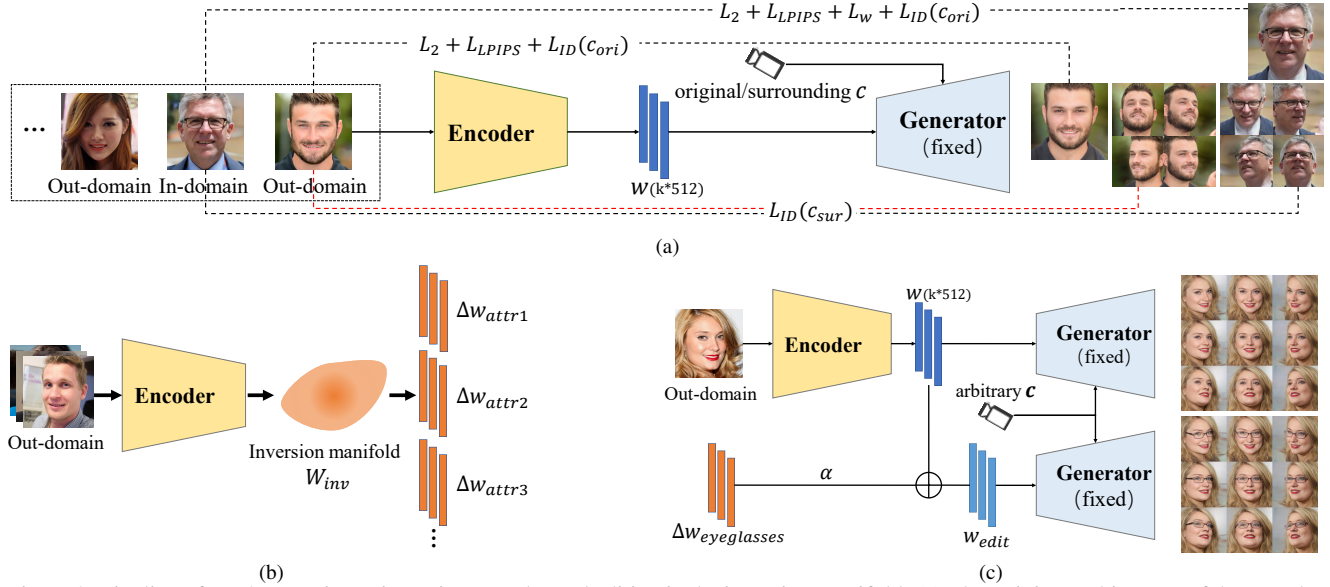


Figure 2. Pipeline of our 3D consistent inversion encoder and editing in the inversion manifold. (a) The training architecture of the encoder, alternating in-domain and out-domain images. $\mathcal{L}_w$ is added in the in-domain round. The multi-view identity loss is calculated between the input image and novel views randomly sampled in the surrounding of the input camera pose. (b) We perform inversions with our encoder on a large real image dataset to produce the inversion manifold $\mathcal{W}_{int}$. Then the editing directions can be found through training an SVM. (c) The inference of 3D GAN inversion and editing. We perform the 3D-aware image editing from a single image conditioned on the desired attribute and arbitrary camera pose.

the representation capacity of the generator [31]. Therefore, in our work, we use the following encoder $E$ to invert the given image to the latent code $w \in \mathcal{W}^+$:

$$
\begin{aligned}
w &= E(X), \\
\hat{X} &= G(E(X), c).
\end{aligned}
\tag{2}
$$

Then the fixed generator $G$ takes $w$ and a given $c$ to produce the inversion image $\hat{X}$.

Editing images is walking along the editing directions in the latent space, which can be linear or non-linear, here we consider linear editing formally given by

$$
X_{edit} = G(w + \alpha \Delta w, c),
\tag{3}
$$

where $\Delta w$ is the editing direction, $\alpha$ is the editing degree.

In the following, we will present how to improve the 3D consistency of the above encoder $E$ and how to find a more precise editing direction. The whole architecture is illustrated in Figure 2.

## 3.2. 3D Consistent Encoder for Inversion

Despite more fidelity near the input camera pose, optimization and hybrid methods will lead to 3D inconsistency of views at large camera poses due to optimization on a single image. The encoder methods alleviate this problem by learning the features of a large number of images with different views. To further improve 3D consistency, we explicitly encourage 3D consistency during training the encoder in two ways: **alternating training scheme** and **multi-view identity loss**, as detailed below.

**Alternating training scheme.** Unlike previous approaches that train the encoder with only real images, we propose an alternating training scheme, which includes in-domain iteration and out-domain iteration. The model takes in a batch of out-domain images, followed by a batch of in-domain images. The encoder is optimized to reconstruct the input images when inverting the out-domain images. When inverting the in-domain images, we additionally optimize the encoder to reconstruct the latent because we have the ground truth. The additional latent code regularization term

is

$$\mathcal{L}_w = \mathbb{E}_w \left[ \|w - E(G(w,c))\|_2^2 \right], \qquad (4)$$

where $G$ is the pretrained generator and $E$ is the encoder.

Alternating training scheme brings two benefits: (i) The training dataset is augmented, increasing the diversity of contents and poses seen by the model; (ii) The regression of the ground latent code encourages the distribution of the inverted latent code closer to the distribution of the original latent code of the generator.

**Multi-view identity loss.** For precise face image inversion and editing, it is challenging to preserve the identity of the input subject. To tackle this, we impose a specific identity loss [28], defined by

$$\mathcal{L}_{ID}(X,c) = 1 - \langle F(X), F(G(E(X),c)) \rangle, \qquad (5)$$

where $F$ is the pretrained ArcFace [10] network which extracts the feature of face, $\langle \cdot \rangle$ is the cosine similarity. $\mathcal{L}_{ID}(X,c)$ denotes the identity loss between the input image $X$ and the image generated by the inversion latent code of the image and the given camera pose $c$.

To improve the identity similarity between images with different poses, we propose a novel loss $\mathcal{L}_{multiID}$ including two terms: the identity loss at the original camera pose and the average identity loss at the surrounding camera pose. It is defined by

$$\mathcal{L}_{multiID} = \lambda_{ori}\mathcal{L}_{ID}(X,c_{ori}) + \\ \lambda_{sur}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{ID}(X,c_{sur}^i), \qquad (6)$$

where $c_{ori}$ are the camera parameters of the input view, $c_{sur}^i$ are the camera parameters of the views surrounding the input view, $N$ is the number of surrounding views sampled, $\lambda_{ori}$ and $\lambda_{sur}$ are the weights of each loss term, respectively. Different sampling strategies can be used to sample the surrounding views centered on the input view. In this paper, we uniformly sample $N = 4$ views from yaw angles between $[-20°, 20°]$ and pitch angles between $[-5°, 5°]$ for an input image.

### 3.3. Total Losses

Our encoder can be trained with natural images in an end-to-end manner. We calculate the commonly used $\mathcal{L}_2$ and $\mathcal{L}_{LPIPS}$ [39] losses between the input image $X$ and the inversion image $\hat{X}$ with the input camera pose to improve the pixel-wise and perceptual similarities.

The total loss is defined as a weight aggregation of all the losses above:

$$\mathcal{L} = \lambda_{l2}\mathcal{L}_2 + \lambda_{lpips}\mathcal{L}_{LPIPS} + \lambda_w\mathcal{L}_w + \mathcal{L}_{multiID}, \qquad (7)$$

where $\lambda_{l2}$, $\lambda_{lpips}$, and $\lambda_w$ are the weights of each loss term, respectively. Note that $\mathcal{L}_{multiID}$ is only used in the human
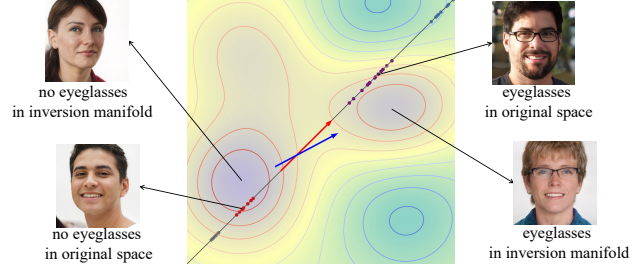


Figure 3. The distortion between the original latent space and the inversion manifold for 1-dimensional $\mathcal{W}$ and $k = 2$. The points in the diagonal line are sampled from the original latent space $\mathcal{W}_{ori}$, which here is represented as a 1-dimensional Gaussian mixture distribution. The entire region represents $\mathcal{W}^2$ space. The warmer cluster corresponds to a higher density of inversion manifold $\mathcal{W}_{inv}$. The red arrow is the editing direction from no eyeglasses in $\mathcal{W}_{ori}$ to eyeglasses in $\mathcal{W}_{ori}$. The blue arrow is the editing direction from no eyeglasses in $\mathcal{W}_{inv}$ to eyeglasses in $\mathcal{W}_{inv}$.

face domain. We simply perform a grid search on these weights to guide the model to produce high-fidelity reconstruction results.

### 3.4. Image Editing in the Inversion Manifold

The general approach to attribute-conditional latent code editing is to find a semantic editing direction in the latent space to change the binary labeled attribute (e.g., young $\leftrightarrow$ old, or no smile $\leftrightarrow$ smile) [14, 22, 28, 31, 32, 35, 36]. Take space $\mathcal{W}^+$ as an example, we sample a latent code $w \in \mathcal{W}^+$, where $\mathcal{W}^+ \subseteq \mathbb{R}^{k*512}$. Formally, we are seeking an editing direction $\Delta w \in \mathcal{W}^+$ such that $w_{edit} = w + \alpha\Delta w$. $\alpha > 0$ will make the edited image look more positive on the attribute, and $\alpha < 0$ represents more negative.

To edit the real image, we need to perform inversion as described in Section 3.1 to obtain the latent code of the image. Here, we have multiple candidates for latent space such as $\mathcal{Z}$, $\mathcal{W}$, $\mathcal{W}^+$, and $\mathcal{S}$ [37]. Strictly speaking, these spaces are manifolds on the corresponding linear space in mathematical terms. However, note that for consistency and simplicity, we still refer to them as space. As described in Section 3.1, we can sample latent codes in these spaces and use them as input to the generator to synthesize the in-domain images. We refer to these spaces used for in-domain images as the original latent spaces of the generator, e.g. the $\mathcal{W}^+$ space is denoted as $\mathcal{W}^+_{origin}$. Instead, we propose that the space consisting of latent codes obtained by inverting a large number of real images via an inversion encoder is called *inversion manifold*. Each original latent space has a corresponding inversion manifold, e.g. the $\mathcal{W}^+_{origin}$ corresponds to $\mathcal{W}^+_{inv}$. To avoid confusion, we call it *inversion manifold* instead of *inversion space*.

It's a non-trivial task to invert a real image to latent code because the generator can not fully model the true distribu-

Figure 4. Qualitative comparison on face inversion and editing at multiple camera poses.

tion [31]. Furthermore, it turns out that there is a distortion between $\mathcal{W}_{inv}^+$ and $\mathcal{W}_{origin}^+$ because the $w'$ obtained by the inversion of the in-domain image generated from $w$ is not equal to $w$. Current popular editing techniques such as InterfaceGAN [31], GANSPace [14], StyleFlow [37], etc., all learn the editing directions in the original latent space. Therefore, editing the latent code in $\mathcal{W}_{inv}^+$ with the editing direction found in $\mathcal{W}_{origin}^+$ leads to distortion. The distortion can be described as:

$$d(\Delta w) = \Delta w_{inv} - \Delta w_{origin}, \qquad (8)$$

where $\Delta w_{origin}$ is the editing direction found in $\mathcal{W}_{origin}^+$, $\Delta w_{inv}$ is the editing direction found in $\mathcal{W}_{inv}^+$. While $\Delta w_{origin}$ can be used to edit the in-domain image very

well, it will lead to imprecision when editing the real image on the inversion manifold by using $\Delta w_{origin}$. We present an illustration of the distortion between $\Delta w_{origin}$ and $\Delta w_{inv}$ in Figure 3. Our method is orthogonal to existing editing techniques, employing them to find editing directions on the inversion manifold. We show the differences between the results between editing in the original space and inversion manifold in Table 5 and Figure 6.

## 4. Experiments

### 4.1. Experimental Settings

**Datasets.** For the human face domain, we train the inversion encoder with FFHQ [19] dataset cropped as [6] and use

| Method | ID | $ID_{20-30}$ | APD | Time(s) |
|---|---|---|---|---|
| IDE-3D [33] | 0.475 | 0.397 | 0.00139 | 277.5 |
| 3D-Inv [21] | 0.476 | 0.457 | 0.00136 | 238.3 |
| Pixel2NeRF [5] | 0.395 | 0.379 | 0.00453 | 0.5 |
| PREIM3D (Ours) | **0.606** | **0.576** | **0.00117** | **0.05** |

Table 2. Quantitative evaluation for inversion on faces. $ID_{a-b}$ denotes the mean ArcFace similarity score between the input image and the 20 inverted images uniformly sampled from yaw angles between $[-b°, a°] \cup [a°, b°]$ and pitch angles between $[-20°, 20°]$.

| Metric | Method | Age | Smile | Eyeglasses |
|---|---|---|---|---|
| ID | IDE-3D [33] | 0.344 | 0.427 | 0.346 |
| | 3D-Inv [21] | 0.425 | 0.482 | 0.420 |
| | Pixel2NeRF [5] | 0.219 | 0.324 | 0.262 |
| | PREIM3D (Ours) | **0.557** | **0.614** | **0.531** |
| AA | IDE-3D [33] | 1.35 | 1.41 | 1.52 |
| | 3D-Inv [21] | 1.41 | 1.49 | 1.61 |
| | Pixel2NeRF [5] | 1.42 | 1.47 | 1.44 |
| | PREIM3D (Ours) | **1.51** | **1.54** | **1.62** |
| AD | IDE-3D [33] | 1.04 | 0.56 | 0.74 |
| | 3D-Inv [21] | 0.94 | 0.56 | 0.62 |
| | Pixel2NeRF [5] | 1.23 | 0.57 | 0.79 |
| | PREIM3D (Ours) | **0.82** | **0.49** | **0.50** |

Table 3. Quantitative evaluation for attribute editing on faces. Attribute altering (AA) measures the change of the desired attribut. Attribute dependency (AD) measures the degree of change on other attributes when edit a certain attribute.

CelebA-HQ [18] for evaluation. We augment the datasets with horizontal flips and estimate the camera parameters of the images following [6]. We use InterfaceGAN [31] for finding the attribute editing directions. The implementation details are provided in Appendix A.2.

### 4.2. Evaluation

We compare our method with three state-of-the-art methods for 3D GAN inversion: Pixel2NeRF [5], IDE-3D [33], and 3D-Inv [21]. Note, Pixel2NeRF and our method are both encoder-based methods. IDE-3D is a hybrid method. 3D-Inv is an optimization-based method. In the comparison experiments, we use the official pretrained models and code for both Pixel2NeRF and IDE-3D, and we implement 3D-Inv according to the paper because they do not release the code. The metrics are calculated on the first 300 images from CelebA-HQ. Because most of these images are front views, we uniformly sample 20 views from yaw angles between $[-30°, 30°]$ and pitch angles between $[-20°, 20°]$ for a source image.

**Quantitative Evaluation.** Table 2 provides quantitative comparisons of the 3D GAN inversion performance. We measure multi-view facial identity consistency (ID) with the average ArcFace similarity score [10] between the sampled images and the source image. Pose accuracy is evalu-



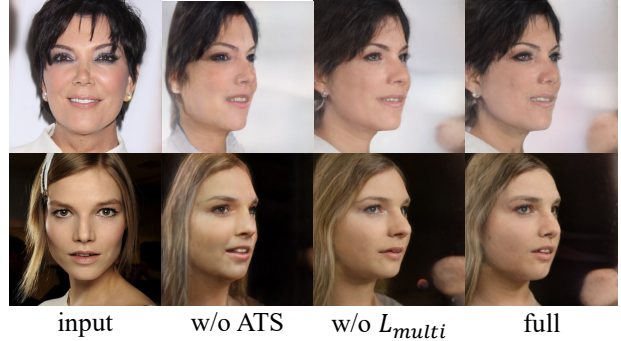input     w/o ATS     w/o $L_{multi}$     full

Figure 5. Inversion results at large camera pose. w/o ATS denotes the model without an alternating training scheme. w/o $\mathcal{L}_{multiID}$ denotes the model without multi-view ID loss.

ated by the average pose distance(APD), which is root mean squared error between the pose encodings estimated by the pretrained 3D face detector [10]. Time metric indicates the average inference time (encoding time and generation time) for one image computed on one Tesla V100 GPU. Our method outperforms baselines on ID and APD metrics and is significantly faster than IDE-3D and 3D-Inv when inference.

We show the comparison of face attribute editing against the baselines in Table 3. We use an off-the-shelf multi-label classifier based on ResNet50 [16] to obtain predicted logits. Attribute altering (AA) measures the change of the desired attribute, which is the attribute logit change $\Delta l_t$ [37] when detecting attribute $t$ by the classifier (pretrained on CelebA [23]). $\Delta l_t$ is normalized by $\sigma(l)$ [37], which is the standard deviation calculated from the logits of CelebA-HQ dataset. We evaluate the precision of attribute editing with attribute dependency (AD) [37], which measures the degree of change on other attributes when modifying along a certain attribute editing direction, as measured by classifiers. Our method performs better than the previous method. More attribute editing results are provided in Appendix.

**Qualitative Evaluation.** We present examples of the inversion and editing results in Figure 4. We sample 9 images for each source image with $yaw = [-30°, 0, 30°]$ and $pitch = [-20°, 0, 20°]$. While optimization-based methods perform better near the input camera pose, optimization on a single image produces artifacts at large camera pose changes, such as head deformation. Compared with previous methods, our method achieves the best 3D consistency, especially at large camera pose. We provide more examples in the Appendix, not only for human faces but also for cats.

### 4.3. Ablation Study

We conduct an ablation study to further validate the benefits of our proposed components and strategies.

**Alternating training scheme.** As analyzed before, the al-

| | $ID_{0-10}$ | $ID_{10-20}$ | $ID_{20-30}$ | $ID_r$ |
|---|---|---|---|---|
| w/o ATS | 0.517 | 0.507 | 0.482 | 0.503 |
| w/o $\mathcal{L}_{multiID}$ | 0.572 | 0.563 | 0.541 | 0.557 |
| full | 0.629 | 0.611 | 0.576 | 0.606 |

Table 4. Effects of alternating training scheme and multi-view ID loss.

| Method | AA | | AD | |
|---|---|---|---|---|
| | origin | inversion | origin | inversion |
| e4e (2D) [35] | 1.53 | **1.59** | 0.45 | **0.38** |
| PREIM3D (Ours) | 1.49 | **1.56** | 0.81 | **0.60** |

Table 5. Performance comparisons in terms of average AA and AD metrics (age, smile, and eyeglasses) on editing in the original space and inversion manifold.

| | IDE-3D | 3D-Inv | Pixel2NeRF | Origin |
|---|---|---|---|---|
| Inversion | 0.55 | 0.51 | 1.0 | - |
| Editing | 0.65 | 0.89 | 0.98 | 0.70 |

Table 6. The result of our user study. The value represents the rate of Ours > others. Origin indicates editing in the original space using our inversion encoder.



Figure 6. Visual comparison between editing in the original space and editing in the inversion manifold.

ternating training scheme and the ground latent code regression encourage the inverted latent space to match the original latent space of the generator. We believe that this will better maintain the 3D consistency of the 3D generator. To validate the effectiveness of this strategy, we show the inversion results in Figure 5. Without the alternating training scheme and the ground latent code regression, the model will lead to significant 3D inconsistency.

**Multi-view ID Loss.** For the human face domain, the multi-view ID loss explicitly guides the model to preserve the identity of the input subject. Following EG3D [6], we calculate the mean ArcFace similarity score between images of the same inverted face at two random camera pose. Our inversion model scored 0.82, while the pretrained EG3D scored 0.77 as reported in their paper. As shown in Table 4, the model with multi-view ID loss improves the identity consistency score. As shown in Figure 5, the multi-view ID loss will encourage subtle face shape adjustments

**Editing in the Inversion Manifold.** Similar to 3D GAN inversion, we can also perform editing in the inversion manifold in 2D space. To validate the generalization of the inversion manifold, we apply it to the state-of-the-art 2D GAN inversion encoder, e4e [35]. Table 5 shows the improvement of the quantitative metrics. Figure 6 demonstrate editing in the inversion manifold will produce more precise results. For example, the third column of the second row in Figure 6 shows that editing eyeglasses in the original space will increase beard and gray hair.
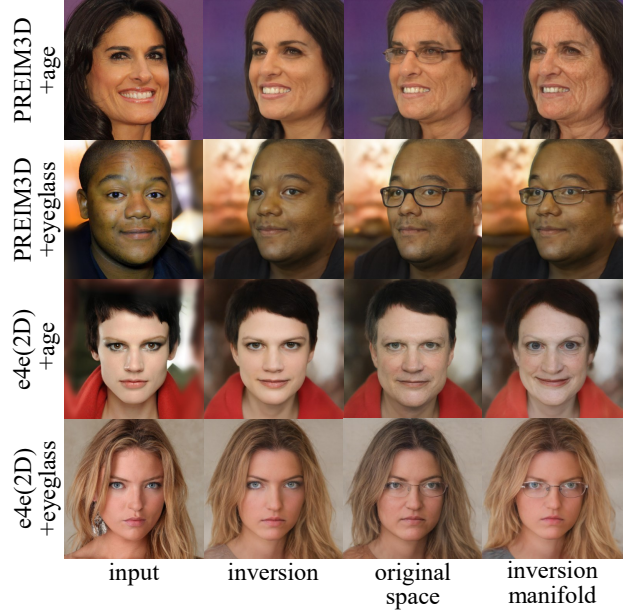
## 4.4. User Study

Considering the human evaluation, we conduct a user study. We collect 1,500 votes from 25 volunteers, who evaluate the 3D consistency and realism of the inversion and editing results. Each volunteer is given a source image, 9 images of our method, and 9 images of baseline (as in Figure 4). According to Table 6, the user study shows our method outperforms the baselines.

## 5. Conclusions

In this paper, we propose a fidelity 3D consistent pipeline that enables 3D reconstruction and 3D-aware editing from a single real image efficiently. With the alternating training scheme, we perform latent code regression to close the gap between the inversion latent code distribution and the original latent code distribution of the generator. This scheme leverages the 3D prior information of the generator and helps to maintain 3D consistency. Benefiting from the multi-view ID loss, our method achieves better identity consistency in the human face domain. We show that editing in the inversion manifold produces more precise results than in the original latent space. Our method can be used for many interactive 3D applications such as virtual reality, metaverse, and avatar-based communication.

**Limitations.** One limitation of our work is the difficulty in dealing with uncommon cases such as delicate earrings and special hairstyles. As our inversion encoder relies on the capacity of the generator to capture real-world scenes, some details were reconstructed imperfectly.

# References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 1, 2, 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8296–8305, 2020. 1, 3

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021. 3

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

[5] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3981–3990, 2022. 2, 3, 4, 7

[6] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 1, 2, 6, 7, 8

[7] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 1, 2

[8] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020. 3

[9] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018. 3

[10] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Cotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5, 7

[11] Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 11379–11388. IEEE, 2022. 1

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

[13] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 1, 2

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020. 1, 2, 3, 5, 6

[15] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11337–11346, 2022. 1, 3

[16] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1510–1519. IEEE Computer Society, 2017. 7

[17] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019. 2, 3

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations, ICLR*, 2018. 7

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 6

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 3

[21] Connor Z Lin, David B Lindell, Eric R Chan, and Gordon Wetzstein. 3d gan inversion for controllable portrait image animation. *arXiv preprint arXiv:2203.13441*, 2022. 1, 2, 3, 4, 7

[22] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *Advances in Neural Information Processing Systems*, 34:16331–16345, 2021. 1, 5

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 7

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2

[25] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 1, 2

[26] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 2

[27] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13503–13513, 2022. 1

[28] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 1, 3, 5

[29] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022. 2, 3

[30] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 1, 2

[31] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 2, 3, 4, 5, 6, 7

[32] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021. 1, 3, 5

[33] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *ACM Transactions on Graphics (TOG)*, 41(6):1–10, 2022. 1, 2, 3, 4, 7

[34] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020. 2, 3

[35] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 1, 3, 5, 8

[36] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022. 1, 3, 5

[37] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12863–12872, 2021. 2, 3, 5, 6, 7

[38] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 3

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 3, 5

[40] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 1, 2