LEARNING A 3D-AWARE ENCODER FOR STYLE-BASED GENERATIVE RADIANCE FIELD

Anonymous authors
Paper under double-blind review

ABSTRACT

We tackle the task of GAN inversion for 3D generative radiance field, (*e.g.*, StyleNeRF). In the inversion task, we aim to learn an inversion function to project an input image to the latent space of a generator and then synthesize novel views of the original image based on the latent code. Compared with GAN inversion for 2D generative models, 3D inversion not only needs to 1) preserve the identity of the input image, but also 2) ensure 3D consistency in generated novel views. This requires the latent code obtained from the single view image to be invariant across multiple views. To address this new challenge, we propose a two-stage encoder for 3D generative NeRF inversion. In the first stage, we introduce a base encoder that converts the input image to a latent code. To ensure the latent code can be used to synthesize identity preserving and 3D consistent novel view images, we utilize identity contrastive learning to train the base encoder. Since collecting real-world multi-view images of the same identity is expensive, we leverage multi-view images synthesized by the generator itself for contrastive learning. Second, to better preserve the identity of the input image, we introduce a residual encoder to refine the latent code and add finer details to the output image. Through extensive experiments, we demonstrate that our proposed two-stage encoder qualitatively and quantitatively exhibits superiority over the existing encoders for GAN inversion in both image reconstruction and novel-view rendering.

1 INTRODUCTION

We aim at tackling GAN inversion of 3D style-based generative radiance fields, which typically combine neural radiance field (NeRF) (Mildenhall et al., 2020) with the generative adversarial network (GAN) (Goodfellow et al., 2014). GAN inversion ((Zhu et al., 2016)) learns a mapping function to project an image into the GAN's latent space. Currently, GAN inversion has been successfully explored in StyleGANs (Karras et al., 2019; 2020b) (*e.g.,* StyleGANv2) which has been used for image synthesis, and enables flexible control of the latent space. Several approaches of GAN inversion are capable of inverting the input image into the latent space (*i.e.*, $\mathcal{W}$ space) (Jahanian et al., 2020; Shen et al., 2020; Tewari et al., 2020; Härkönen et al., 2020) or extended latent space (*i.e.*, $\mathcal{W}+$ space: concatenation of all $\mathcal{W}$ latent code from each layer) (Abdal et al., 2019; 2020; Zhu et al., 2020; Abdal et al., 2021) for image editing. However, the exploration of GAN inversion is currently limited to 2D-based GAN and few works have studied the encoder-based GAN inversion of 3D style-based generative models, as shown in Figure 1.

Recently, 3D style-based generative models using radiance field (*i.e.*, NeRF (Mildenhall et al., 2020)), such as EG3D (Chan et al., 2022) or StyleNeRF (Gu et al., 2021), have been proposed for unsupervised generation of multi-view consistent images. Similar to StyleGANs, these NeRFs learn a controllable $\mathcal{W}$ space and enable explicit 3D camera control, using only single-view 2D training images. To achieve 3D style-based GAN inversion for these models, one straightforward way is to directly apply the aforementioned 2D inversion methods, by feeding the 2D image and the corresponding camera poses as inputs, and rendering the corresponding multi-view images (see Figure 1b). However, there are two main challenges. First, if only existing single-view images are used to train the inversion method, the predicted latent code only works when generating images of the same view (camera pose), but fails to generate the high-quality image for novel views, see Figure 1 (c). To address this issue, we need multi-view images to train the inversion method. However, this leads to our second challenge: it may be difficult, if not infeasible, to collect sufficient

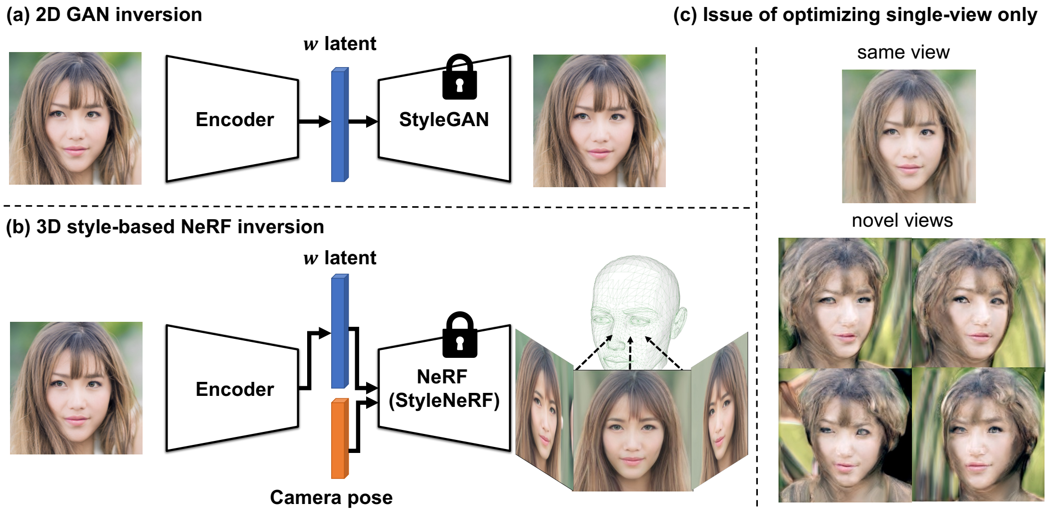

Figure 1: **Comparison of encoder-based GAN inversion between (a) 2D GAN and (b) 3D generative NeRF**. Compared with 2D GAN, it is challenging to perform 3D generative NeRF inversion from the single view image since the latent is assumed to be view-invariant. (c) Directly applying the existing effective 2D inversion approaches (pSp and ReStyle) would lead to distribution misalignment in W+ space.

multi-view images with known camera poses for training. For example, we may need calibrated and synchronized camera arrays to collect such images.

To learn an inversion function without the use of multi-view images, we propose a framework named 3DE-NeRF to learn a 3D-aware Encoder for style-based NeRFs. 3DE-NeRF is composed of a two-stage learnable encoder: a base encoder and a residual encoder. First, we introduce a base encoder to learn the view-invariant latent code in $\mathcal{W}$ space based on the observation that $\mathcal{W}+$ is prone to spoil 3D structure (see Figure 5). Moreover, we leverage synthesized images (*i.e.*, using the multi-view images generated by the model itself) and contrastive learning with the triplet loss to learn a better view-invariant latent code in $\mathcal{W}$. Second, since the latent code in $\mathcal{W}$ space is known to be more difficult to fully reconstruct the input image compared with $\mathcal{W}+$ (Karras et al., 2020b), we propose a residual encoder to refine the latent code from the base encoder in $\mathcal{W}+$ space. It adds more fine-grained details to the generated image, which makes it more consistent with the input image. More specifically, the residual encoder first takes the concatenation of the generated image from the base encoder and the input image as inputs and then produces the refined latent residue in $\mathcal{W}+$ space. The proposed framework allows us to not only learn the view-invariant latent code but to also preserve the fine-grained details in the final generated image that matches the input image.

We verify the effectiveness of the proposed method and its key components using StyleNeRF (Gu et al., 2021) as the pre-trained generator for GAN inversion. Moreover, to test the generalization ability of the proposed method, we combine it with the online optimization technique PTI (Roich et al., 2021) as well as apply it to a different pretrained generator EG3D (Chan et al., 2022) (see appendix for details). The contributions of this paper are summarized as follows:

- We demonstrate the challenges of GAN inversion for 3D style-based NeRF and the limitations of the current 2D encoder-based models for this task.
- We propose an encoder-based framework named 3DE-NeRF, which consists of a base encoder and a residual encoder, to perform GAN inversion for the 3D generative neural radiance field.
- Compared with the existing encoders for GAN inversion, our proposed model achieves more effective GAN inversion for the 3D generative NeRF and has superior image quality for rendering novel views.
- Our proposed framework has good generalization to enable online optimization methods (*e.g.* PTI) to render novel views and to invert more 3D style-based generators (*e.g.* EG3D).

2 Related Works

Generative Adversarial Network. GANs (Goodfellow et al., 2014) have demonstrated success in image synthesis and have been extended to a number of works (Zhang et al., 2019; Brock et al., 2018; Karras et al., 2018). StyleGANs (*e.g.,* StyleGAN (Karras et al., 2019), StyleGAN2 (Karras et al., 2020b), and StyleGAN2-ada (Karras et al., 2020a)) achieve state-of-the-art image quality and support different levels of semantic manipulation. In particular, many methods have been proposed for finding these semantic latent space manipulation using varying levels of supervision. These include full-supervision in the form of semantic labels (Abdal et al., 2021; Shen et al., 2020; Goetschalckx et al., 2019) and unsupervised approaches (Wang & Ponce, 2021; Voynov & Babenko, 2020). Some methods (Härkönen et al., 2020; Tewari et al., 2020; Abdal et al., 2020; Shoshan et al., 2021) also leverage disentangled properties in the latent space to enable 3D controls. However, most of these works focus on the rendering of 2D images with 3D controls and are not capable of manipulating camera poses easily as volumetric rendering (NeRF (Mildenhall et al., 2020)).

Image Synthesis with Generative NeRF. Methods built on implicit functions, e.g., NeRF (Mildenhall et al., 2020), have been proposed in (Chan et al., 2021; Schwarz et al., 2020; Pan et al., 2021). To generate high-resolution images conditioned on the input style latent code, EG3D (Chan et al., 2022), StyleNeRF (Gu et al., 2021), VolumeGAN (Xu et al., 2022), StyleSDF (Or-El et al., 2022), and GMPI (Zhao et al., 2022) have been developed to generate multi-view images with latent and pose control. In addition, some works such as Sofgan (Chen et al., 2022a) and Sem2NeRF (Chen et al., 2022b) are able to perform multi-view synthesis with NeRF by taking into multi-view or single-view semantic masks. Among these models, StyleNeRF (Gu et al., 2021) is able to perform novel-view image synthesis given the style latent code and the camera pose, and is only relies on MLP layers as the classical NeRF (Mildenhall et al., 2020). To simplify the analysis of GAN inversion of 3D style-based NeRF, we employ StyleNeRF in our experiments.

GAN inversion. GAN inversion (Zhu et al., 2016) is the process of obtaining a latent code that can allow the generator to reconstruct the given image. Generally, inversion methods either directly optimize the latent vector to minimize the loss for a given image (Abdal et al., 2019; 2020; Bau et al., 2020; Gu et al., 2020), train an encoder on a large number of images to learn a mapping from an image to a style latent (Alaluf et al., 2021; Guan et al., 2020; Kang et al., 2021; Kim et al., 2021; Pidhorskyi et al., 2020; Richardson et al., 2021; Tov et al., 2021; Wang et al., 2022), or use a hybrid approach leveraging both methods (Zhu et al., 2016; 2020). For the encoder-based methods, pSp (Richardson et al., 2021) proposes a feature pyramid encoder into $\mathcal{W}+$ space. ReStyle (Abdal et al., 2019) iteratively refines the predicted style latent through a few forward passes. However, these effective approaches are designed for 2D StyleGAN. Recently, IDE-3D (Sun et al., 2022) propose inversion approach for 3D neural renderer with semantic masks, which can not generalize to several pre-trained style-based NeRFs. Here, we would like to design an inversion model which is generalizable for 3D style-based NeRF inversion.

3 The Proposed Approach

3.1 Problem Formulation and Overview

Inversion of 2D generative model: In the encoder-based 2D GAN inversion, the goal is to train an encoder E to generate the latent code w[1] for the given target image x and minimize the distance between the input image and the generated image:

$$\min_{E} \mathcal{L}(x, G(w)), \text{ s.t. } w = E(x) \tag{1}$$

where G indicates the generator (*i.e.,* StyleGAN). The objective can be L_2 distance, perceptual distance (LPIPS) (Zhang et al., 2018), or a more sophisticated loss which consists of various reconstruction losses and regularization terms. We use an encoder to compute a latent code $w = E(x)$ to minimize the re-construction loss. This allows fast inference without per-input optimization.

[1] latent $w \in \mathcal{W}$ (Shen et al., 2020) or the extended latent $w \in \mathcal{W}+$ (Karras et al., 2020b)

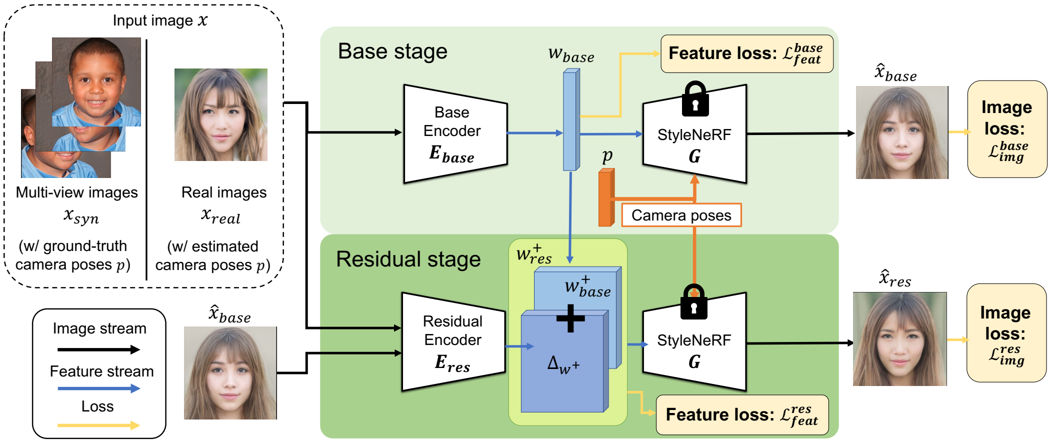
Input image x
Multi-view images x_{syn}
(w/ ground-truth camera poses p)
Real images x_{real}
(w/ estimated camera poses p)
Image stream
Feature stream
Loss
$\hat{x}_{base}$
Base stage
Base Encoder E_{base}
w_{base}
Feature loss: $\mathcal{L}^{base}_{feat}$
StyleNeRF G
p
Camera poses
$\hat{x}_{base}$
Image loss: $\mathcal{L}^{base}_{img}$
Residual stage
Residual Encoder E_{res}
w^+_{res}
w^+_{base}
Δ_{w^+}
StyleNeRF G
$\hat{x}_{res}$
Image loss: $\mathcal{L}^{res}_{img}$
Feature loss: $\mathcal{L}^{res}_{feat}$

Figure 2: **Overview of our proposed 3D-aware style-based NeRF Encoder: 3DE-NeRF.** It consists of two stages: base stage and residual stage, and is trained with feature-level 3D-aware losses [2]and image-level reconstruction losses. More details can be referred to the section 3.

Inversion of 3D generative model: For 3D style-based NeRF inversion, we not only need to reconstruct same-view images but also generate novel views of the same identity:

$$\min_{E} \sum_{i=0}^{n} \mathcal{L}(x_i, G(w, p_i)), \text{ s.t. } w = E(x_0), \tag{2}$$

where x_i, $i = 0, 1, .., n$ represent multi-view images that has the same identity as x_0 and p_i are the corresponding camera poses. Minimizing the objective allows the model to learn the view-invariant latent code $\hat{w}$ since it maps multi-view images x_i (controlled by the pose p_i) to the same latent code w for each set of the training sample. During inference, a single-view image is mapped to the latent code $\hat{w}$ which can produce multi-view images of the same identity by changing the poses.

Method overview: In order to perform GAN inversion for style-based generative NeRFs, we propose an encoder-based framework named 3DE-NeRF, and the overview of the pipeline is presented in Figure 2. The 3DE-NeRF involves two stages: the base stage and the residual stage. 1) In base stage, the introduced base encoder E_{base} takes an image x as input and produces the style latent code w_{base}. In order to learn the view-invariant latent code, we leverage the multi-view images synthesized by the generator, shown in Figure 3. This latent code w_{base} is optimized to roughly reconstruct the 2D input image $\hat{x}_{base}$ and enable 3D-consistent novel-view rendering. 2) To further minimize the identity gap between the output and the input images, a residual encoder E_{res} is introduced to refine the latent code w_{base}. It first takes the concatenation of the input image x and the generated image $\hat{x}_{base}$ from the previous stage as input and learns a residue Δ_{w^+}. Then we can obtain the output style latent code $\hat{w} = w^+_{res}$ by adding the residue to w^+_{base}.

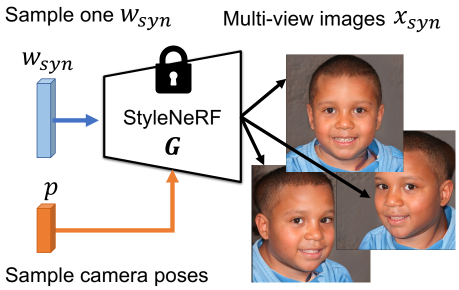

Figure 3: The process of generating multi-view images by feeding the same latency code with different camera poses to StyleNeRF.

3.2 Preliminary of Style-based NeRF (StyleNeRF)

Style-based Neural Radiance Field. Following StyleGANs (Karras et al., 2019; 2020b), StyleNeRF (Gu et al., 2021) also introduce the mapping network f which maps noise vectors from a spherical Gaussian space $\mathcal{Z}$ to the style space $\mathcal{W}$. f consists of several MLP layers and the input style vector $w \in \mathcal{W}$ can be derived by $w = f(z), z \in \mathcal{Z}$. Following the neural rendering mechanism

[2] Feature-level losses include triplet losses and L1 losses, which can only be applied to synthesized multi-view images in this work.

in NeRF (Mildenhall et al., 2020), our model also takes the position $u \in \mathbb{R}^3$ and viewing direction $d \in \mathbb{S}^2$ as inputs, and predicts the density $\sigma(u) \in \mathbb{R}$ and view-dependent color $c(u, d) \in \mathbb{R}^3$.

In order to render the color and density for each coordinate in 3D space with high-frequency details, StyleNeRF also uses positional embedding with Fourier series: $\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))$, where the function $\gamma(.)$ is applied to each of the three coordinates in u and to the three coordinates of the view direction d. Let us denote the rendering network by ϕ_w^n where n indicates the number of MLP layers within and w indicates the style feature. Each MLP weight matrix is modulated by the latent code w independently. Both the density and the color can be rendered respectively with:

$$\sigma_w(x) = h_\sigma(\phi_w^n(\gamma(u))) \qquad c_w(u, d) = h_c(\phi_w^n(\gamma(u)), \gamma(d)), \tag{3}$$

where $h_\sigma(\cdot)$ and $h_c(\cdot)$ are projection layers.

Volume Rendering with Radiance Fields. Once we have the color and density for each coordinate and view direction, we render the color $C(r)$ for each pixel along that camera ray $r(t) = o + td$ passing through the camera center o with volume rendering (Kajiya & Von Herzen, 1984):

$$C_w(r) = \int_{t_n}^{t_f} T(t)\sigma_w(r(t))c_w(r(t), d)dt, \text{where} \quad T(t) = \exp(-\int_{t_n}^{t} \sigma_w(r(s))ds). \tag{4}$$

The function $T(t)$ denotes the accumulated transmittance along the ray from t_n to t. In practice, the continuous integration is discretized by accumulating sampled points along the ray. More details can be obtained in NeRF (Mildenhall et al., 2020) and StyleNeRF (Gu et al., 2021).

3.3 Inversion of the view-invariant latent in W

Unlike 2D GAN inversion which only generates the output image with the same camera pose as input image, the inversion of 3D generative NeRF has to consider the optimization of unseen views for the input image. However, the latent code $w \in \mathcal{W}$ obtained by training on single-view images may not lead to high-quality novel-view images. See Figure 1 (c).

In order to learn the 3D-aware latent code w, we introduce a base encoder E_{base} that is able to generate view-invariant latent code. In other words, for multi-view images of the same identity, we hope E_{base} to map them to the same latent code: $w_{base}^{(i)} = E_{base}(x_j^{(i)})$, where $x_j^{(i)}$ denotes an image corresponding to camera pose p_j of the identity-i. To ensure this, we can use contrastive learning to train the encoder. Specifically, we perform contrastive learning with triplet loss $\mathcal{L}_{tri}$ on the feature vector w, which would maximize the inter-class discrepancy while minimizing intra-class distinctness. Specifically, for each input image x, we sample a positive image x_{pos} with the same identity label and a negative image x_{neg} with different identity labels to form a triplet tuple. Then, the following equations compute the distances between x and x_{pos}/x_{neg}:

$$d_{\text{pos}} = \|w_{base} - w_{base_{\text{pos}}}\|_2, \quad d_{\text{neg}} = \|w_{base} - w_{base_{\text{neg}}}\|_2, \tag{5}$$

where w_{base}, $w_{base_{\text{pos}}}$, and $w_{base_{\text{neg}}}$ represent the feature vectors of images x, x_{pos}, and x_{neg}, respectively. With the above definitions, we have the triplet loss $\mathcal{L}_{tri}$ defined as

$$\mathcal{L}_{tri}(w_{base}) \ = \max(0, m + d_{\text{pos}} - d_{\text{neg}}), \tag{6}$$

where $m > 0$ is the margin used to define the distance difference between the positive image pair d_{pos} and the negative image pair d_{neg}. Contrastive learning requires multi-view images with the same identities. In reality, collecting such datasets is nontrivial, as it requires synchronized and calibrated camera arrays. To bypass this, we utilize images synthesized by the generator (*i.e.,* StyleNeRF) itself. We can sample latent codes w_{syn} from a StyleNeRF's $\mathcal{W}$ space, then sample different camera poses to generate multi-view images of the same identities as x_{syn}.

Since we have the w_{syn} latent code for x_{syn}, we can directly apply an L_1 loss between the predicted w_{base} and the "ground-truth" w_{syn}. The feature-level loss for synthesized images is summed up as:

$$\mathcal{L}_{feat}^{base} \ = \mathcal{L}_{tri}(w_{base}) + \mathcal{L}_1(w_{base}, w_{syn}), \tag{7}$$

On the other hand, we are able to utilize both the real images x_{real} and synthesized images x_{syn} to train the base encoder with image-level loss. We construct the image-level loss using the pixel-wise L_2 loss and LPIPS loss (Zhang et al., 2018). Following pSp (Richardson et al., 2021), we also apply an identity (ID) similarity loss by employing a pre-trained facial recognition ResNet-IRSE50 (Deng et al., 2019) to measure the facial identity:

$$\mathcal{L}_{img}^{base} = \mathcal{L}_2(\hat{x}_{base}, x) + \mathcal{L}_{\text{LPIPS}}(\hat{x}_{base}, x) + \mathcal{L}_{ID}(\hat{x}_{base}, x), \tag{8}$$

where $\hat{x}_{base} = G(E_{base}(x), p)$ and p indicates the corresponding camera pose. We use the ground truth camera poses for synthesized images and camera poses predicted by the off-the-shelf predictor (Ruiz et al., 2018) for real images. The image-level losses can be summed up as for both synthesized images and real images. With the image-level loss $\mathcal{L}_{img}^{base}$, the base encoder is able to learn to reconstruct the synthesized and the real images by back-propagating through the generator.

3.4 Refinement of the latent in W+

While w latent code is learned to preserve the 3D structure with our base encoder E_{base}, it leads to poor identity preservation. Thus, we introduce a residual encoder to refine the latent code w_{base}.

Following previous works for learning the $\mathcal{W}+$ latent instead of $\mathcal{W}$, we first duplicate the base latent $w_{base} \in \mathbb{R}^d$ to $w_{base}^+ \in \mathbb{R}^{n \times d}$ and learn the refined w_{res}^+ by adding the learned residue Δ_{w^+}:

$$w_{res}^+ = w_{base}^+ + \Delta_{w^+} \quad \text{and} \quad \Delta_{w^+} = E_{res}(x, \hat{x}_{base}), \tag{9}$$

where w_{res}^+ is in the $\mathcal{W}+$ latent space and is capable of better reconstructing the input image using the generator G. In order to learn the residue Δ_{w^+}, we introduce the residual encoder E_{res} which takes the input image x and the previously generated image $\hat{x}_{base}$ as inputs and produces Δ_{w^+}. The design of the residual stage is similar to the ReStyle (Alaluf et al., 2021) originally proposed for 2D StyleGAN. The difference lies in that Restyle (Alaluf et al., 2021) uses the randomly averaged w latent code and the corresponding synthesized image as inputs while our E_{res} uses the outputs (i,e, w_{base} and x_{base}) of the base encoder. Same as the base stage, we employ the same image-level losses to train the residual encoder:

$$\mathcal{L}_{img}^{res} = \mathcal{L}_2(\hat{x}_{res}, x) + \mathcal{L}_{\text{LPIPS}}(\hat{x}_{res}, x) + \mathcal{L}_{ID}(\hat{x}_{res}, x), \tag{10}$$

where $\hat{x}_{res} = G(E_{res}(x, \hat{x}_{base}), p)$ and p indicate the corresponding camera poses similar to base stage. x can either be a real image or a synthetic image. Since we also have ground truth w_{syn}^+ from synthesized images, we can also train the residual encoder using the feature-level L_1 loss:

$$\mathcal{L}_{feat}^{res} = \mathcal{L}_1(w_{res}^+, w_{syn}^+), \tag{11}$$

Note that we do not use the triplet loss on the residual latent code w_{res}^+ since the large dimension size of the $\mathcal{W}+$ space ($\mathbb{R}^{n \times d}$) makes the contrastive learning prone to overfitting.

Similar to Restyle (Alaluf et al., 2021), our residual encoder can also perform multiple iterative refinement using the Equation 9.

The total loss $\mathcal{L}$ for training our proposed 3DE-NeRF is summarized as follows:

$$\mathcal{L}_{total} = \lambda_{feat}^{base} \cdot \mathcal{L}_{feat}^{base} + \lambda_{img}^{base} \cdot \mathcal{L}_{img}^{base} + \lambda_{feat}^{res} \cdot \mathcal{L}_{feat}^{res} + \lambda_{img}^{res} \cdot \mathcal{L}_{img}^{res}, \tag{12}$$

where λ_{feat}^{base}, λ_{img}^{base}, λ_{feat}^{res} and λ_{img}^{res} are the hyper-parameters used to control the weighting of the corresponding losses.

4 Experiment

4.1 Experimental Settings

Datasets. By default, all our experiments are conducted on human faces and using StyleNeRF (Gu et al., 2021) as the pretrained generator for GAN inversion. We train the encoder for StyleNeRF using real images in **FFHQ** (Karras et al., 2019) (i.e., the same dataset used for StyleNeRF training) and multi-view synthesized images from StyleNeRF itself. We use the **CelebA-HQ** test set (Karras et al., 2018; Liu et al., 2015) for quantitative evaluations. To prevent real human face privacy issues,

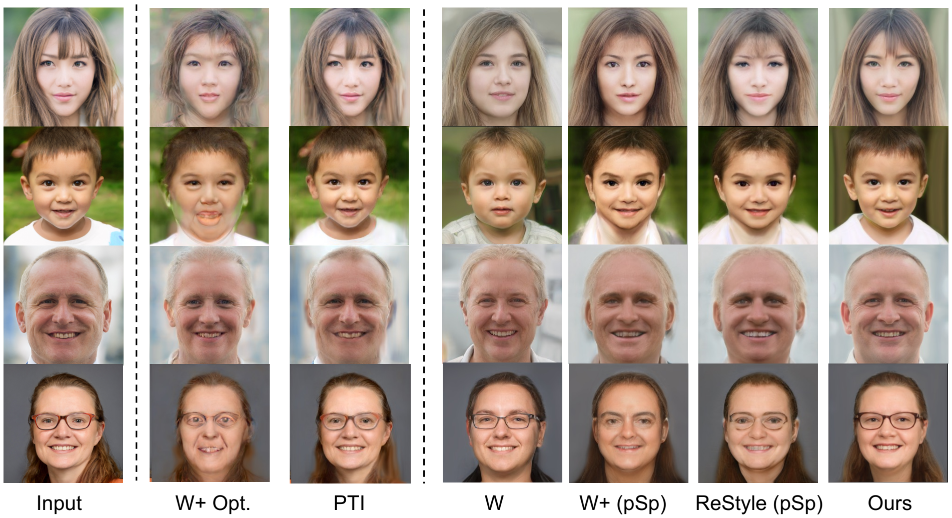

Figure 4: **Qualitative comparisons on image reconstruction.** All of the output images are rendered using the same camera pose as the input image from the StyleGAN2-Fake dataset.

we do not present the qualitative results on real images (CelebA-HQ) but visualize the results on the fake images, **StyleGAN2-Fake**, which has 263 curated images generated and released by StyleGAN2 (Karras et al., 2020b). The camera poses for all real input are derived using the off-the-shelf pose estimator: HopeNet (Ruiz et al., 2018) for a fair comparison with previous works. For synthesized images, we use their ground truth camera poses for both training and inference. Moreover, we will present results on animal faces on AFHQ dataset(Choi et al., 2020) and the results based on EG3D (trained with FFHQ or AFHQ-cat and its self-generated images) in A.5 of the appendix.

Baselines. Since our 3DE-NeRF is the first 3D-aware encoder for generative style-based NeRFs, we compare it with several baselines. The first set of baselines are directly built from current state-of-the-art 2D styleGAN inverters, including pSp (Richardson et al., 2021) and ReStyle (Alaluf et al., 2021). We also build a baseline encoder for $\mathcal{W}$ space inversion. For a fair comparison, all of the encoder-based competitors are trained on the same dataset, *i.e.*, using both real and synthesized images. To compare with online optimization methods, we compare our model with latent vector optimization in $\mathcal{W}+$ (Karras et al., 2020b) and PTI (Roich et al., 2021).

Evaluation settings. We conduct the experiments in two settings: 1) Same-view image reconstruction and 2) Novel-view image rendering. For the first setting, we visually compare the input image and the corresponding output image generated from the latent code and the camera pose of the input image. We also quantitatively evaluate the distance between the input and output images using the metrics: L_2, LPIPS (Zhang et al., 2018), MS-SSIM (Wang et al., 2003), and identity (ID) (Huang et al., 2020). For the second setting, we qualitatively and quantitatively compare the input image and the novel views (e.g., $-35°$ yaw angle) image generated from its latent code. Since we do not have the ground-truth to measure the distance, we only quantitatively evaluate the identity (ID) distance (Huang et al., 2020) with input from different views. We would like to note that, the original generator (StyleNeRF (Gu et al., 2021)) is trained using the head yaw angle ranging between $-17°$ to $17°$ degrees. Based on our observation, the pretrained StyleNeRF itself can only generate images at most twice the yaw range (i.e., $-35° \sim +35°$) before breaking the 3D structure. Thus, we set the rendering yaw range of our 3DE-NeRF to $-35° \sim +35°$ and the default roll angle as $0°$.

4.2 Results of image reconstruction

In this section, we compare our proposed model with three encoder-based models and two optimization approaches quantitatively (in Table 1) and qualitatively (in Figure 4). As listed in Table 1, among all encoder-based methods, our proposed method achieves the best results across all four metrics. For example, it outperforms the 2D StyleGAN encoder, i.e., ReStyle (Or-El et al., 2022), with a large gap and outperforms the other two baselines even more, which demonstrates that it is

	Method	$\downarrow L_2$	$\downarrow$ LPIPS	$\uparrow$ MS-SSIM	$\uparrow$ ID	Time (s)$\downarrow$
Online-based	PTI (Roich et al., 2021)	0.03	0.09	0.86	0.67	194.203
	$\mathcal{W}+$ Opt. (Karras et al., 2020b)	0.08	0.28	0.65	66.153	
Encoder-based	$\mathcal{W}$	0.12	0.31	0.60	0.21	0.105
	$\mathcal{W}+$ (pSp) (Richardson et al., 2021)	0.14	0.29	0.63	0.32	0.132
	Restyle (pSp) (Alaluf et al., 2021)	0.09	0.27	0.68	0.37	0.454
	3DE-NeRF (Ours)	**0.05**	**0.21**	**0.72**	**0.55**	0.315

Table 1: **Quantitative comparison on image reconstruction** with online-based (i.e., upper bound for reconstruction) and encoder-based methods on the CelebA-HQ test dataset.

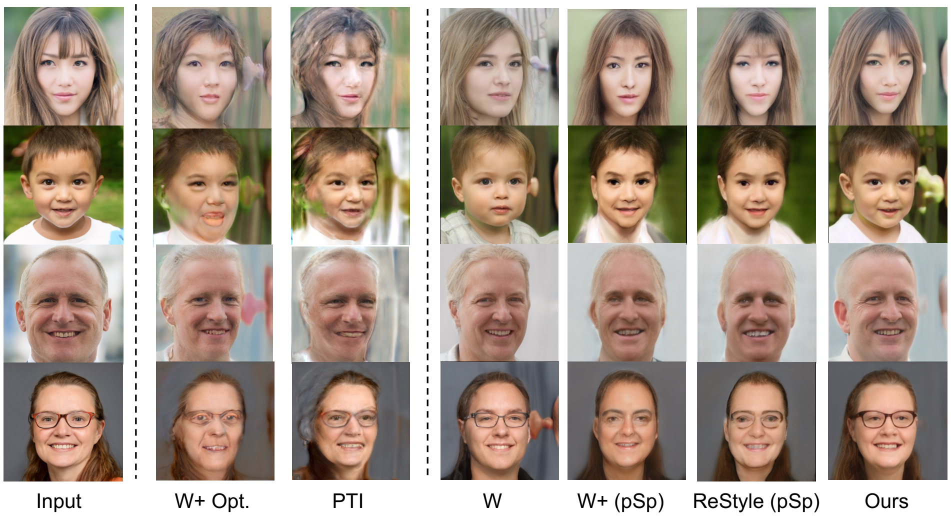

Figure 5: **Qualitative comparisons on novel-view rendering.** All of the output images are rendered using face yaw angle $-35°$ degree. The images are from the StyleGAN2-Fake dataset.

not optimal to directly apply 2D encoders to invert 3D style-based NeRFs. Visually, our model also greatly outperforms all encoder-based methods as shown in Figure 4. For example, although all models can generate realistic faces due to the pretrained StyleNeRF, our proposed model can better reconstruct the input image which is consistent with our quantitative results. To dig deeper, we observe that although encoding in $\mathcal{W}$ achieves the worst image reconstruction results, it has much better 3D preservation than encoding the latent in $\mathcal{W}+$, which we will discuss in the next section (see column 5 & 6 vs column 4 in Figure 4 and Figure 5. This is also what motivates us to build our base encoder to learn latent code in $\mathcal{W}$ space rather than $\mathcal{W}+$ space.

Besides encoder-based baselines, we also compare our proposed model with the 2D online optimization methods (see row 2 &3 in Table 1). The online optimization methods are much slower than encoder-based methods, and in return, their performance for 2D image reconstruction is known to be the upper bound for that of encoder-based methods (Alaluf et al., 2021; Richardson et al., 2021). Our model has a small gap (or even better ID metric) compared with the online method $\mathcal{W}+$ Opt. (see Table 1 row 3) while $\mathcal{W}+$ Opt. does not perfectly reconstruct the input image although it works effectively in 2D StyleGAN. PTI still has the best performance for image reconstruction with 3D style-based NeRF quantitatively and qualitatively. However, we greatly reduced the gap between encoder-based methods and the online optimization method for same-view image reconstruction.

4.3 Results of novel-view rendering

We present our qualitative result of novel viewing rendering of yaw angle $-35°$ degree in Figure 5, and also compare it with the same encoder-based models and optimization approaches. First, comparing with encoder-based models for $\mathcal{W}$, $\mathcal{W}+$ using pSp (Richardson et al., 2021) and using ReStyle (Alaluf et al., 2021), we found that our proposed 3DE-NeRF not only effectively preserves the fine details and identity from the input image, but also maintains a reasonable 3D shape. Second, we observe that while encoding the latent in $\mathcal{W}+$ is more effective than encoding in $\mathcal{W}$ in image reconstruction, it generates inaccurate face angle or loses 3D preservation for novel views

	Method	↑ ID				
		Yaw angle				Avg.
		−35°	−17°	17°	35°	
Online-based	PTI (Roich et al., 2021)	0.41	0.46	0.44	0.40	0.43
	$\mathcal{W}+$ Opt. (Karras et al., 2020b)	0.23	0.28	0.27	0.22	0.25
Encoder-based	$\mathcal{W}$	0.17	0.19	0.19	0.15	0.18
	$\mathcal{W}+$ (pSp) (Richardson et al., 2021)	0.21	0.27	0.31	0.24	0.22
	Restyle (pSp) (Alaluf et al., 2021)	0.20	0.35	0.32	0.21	0.27
	3DE-NeRF (Ours)	**0.49**	**0.53**	**0.53**	**0.50**	**0.51**

Table 2: **Quantitative results on novel-view rendering, and comparison with online-based and encoder-based methods.** The results are measured on the CelebHQ test dataset.

(see column 5 & 6). On the other hand, encoding latent in $\mathcal{W}$, though with much worse identity preservation, has better 3D preservation and correct view-angle. Third, the optimization method PTI (Roich et al., 2021) though has near-perfect image reconstruction of the same view, it breaks 3D structure when rendering novel views (See column 3 of Figure 4 and Figure 5). Consequently, compared with all baselines including online optimization and encoder-based models, our proposed 3DE-NeRF achieves superior results in novel-view rendering.

We also benchmark the quantitative results using identity metrics for the selected four yaw head angles. As shown in Table 2, the score usually decreases when the yaw head pose is more extreme. Our proposed model achieves the highest ID score among all competitors. The optimization method PTI (Roich et al., 2021), though has achieved the highest scores in all of the evaluation metrics in Table 1, also exhibits an inferior ID score to our encoder-based model. To adapt the 2D optimization method to 3D style-based NeRFs, we observe that, for both PTI and $\mathcal{W}+$ Opt approaches, our encoder can improve the identity and 3D preservation, especially for novel-view rendering (see appendix A.3). Results on more novel views rendering for different input images (e.g., different races, gender, age, and skin tones) can be found in appendix A.4, and our demo video.

4.4 Ablation studies

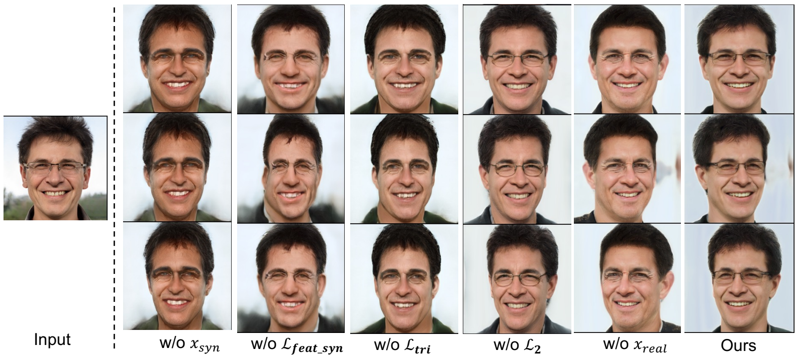

Figure 6: Ablation studies on the types of training images and the feature-level loss.

To further analyze the effectiveness of essential components of the proposed method, we conduct the experiments with one of them excluded and present the qualitative result in Figure 6. When the synthesized images x_{syn} are excluded (note that the feature losses $\mathcal{L}_{feat}$ will also be excluded without x_{syn}), the style latent generated by our proposed encoders could not preserve the reasonable face structure (see column 2). When using synthesized images x_{syn} and only excluding $\mathcal{L}_{feat}$, the generated multi-view images still have artifacts and distortions in the face (see column 3). In addition, we found that $\mathcal{L}_{tri}$ in $\mathcal{L}_{feat}$ serves as a more important role for the 3D and identity preservation (see column 4 vs column 5).Moreover, if the encoders are trained with synthesized images x_{syn} only (*i.e.*, w/o x_{real}), the style latent code is able to preserve view consistency yet still has the loss of identity preservation compared with the full model (see column 6 vs column 7). These studies demonstrate that both real images and synthesized images with feature-level losses are significant to our model.

To further analyze the importance of both our base encoder and the residual encoder, we also visualize the output of these two encoders in appendix A.4. To further verify the generalization ability of the proposed method, the results of using the EG3D generator and AFHQ animal dataset are visualized in appendix A.4 and A.5, respectively.

5 Conclusion

We have unveiled the challenges of GAN inversion for 3D style-based NeRF and the limitations of the current 2D encoder-based models through experiments. To tackle the issue, we propose an encoder-based framework named 3DE-NeRF, which consists of a base encoder and a residual encoder, to perform GAN inversion for the 3D generative radiance field. Compared with the current existing encoder-based methods for GAN inversion, our proposed model achieves more effective GAN inversion for 3D generative NeRF and has satisfactory image quality for rendering novel views. We also demonstrate that the style latent code generated by our proposed model is able to serve as a good initial point for online optimization.

References

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *CVPR*, pp. 4432–4441, 2019.

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, pp. 8296–8305, 2020.

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.

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *ICCV*, pp. 6711–6720, 2021.

David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.*, 2020.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018.

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

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pp. 16123–16133, 2022.

Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *ACM Transactions on Graphics (TOG)*, 41(1):1–26, 2022a.

Yuedong Chen, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Sem2nerf: Converting single-view semantic masks to neural radiance fields. *ECCV*, 2022b.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020.

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

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, pp. 5744–5753, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3012–3021, 2020.

Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *CORR*, 2020.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *NeurIPS*, 33:9841–9850, 2020.

Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *CVPR*, pp. 5901–5910, 2020.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *ICLR*, 2020.

James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984.

Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations. In *ICCV*, pp. 13941–13949, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020b.

Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *CVPR*, pp. 852–861, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pp. 3730–3738, 2015.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pp. 405–421. Springer, 2020.

Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *CVPR*, pp. 13503–13513, 2022.

Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. *NeurIPS*, 34:20002–20013, 2021.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *CVPR*, pp. 14104–14113, 2020.

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 *CVPR*, pp. 2287–2296, 2021.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021.

Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2074–2083, 2018.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *NeurIPS*, 33:20154–20166, 2020.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, pp. 9243–9252, 2020.

Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gerard Medioni. Gan-control: Explicitly controllable gans. In *ICCV*, pp. 14083–14093, 2021.

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. *Siggraph Asia*, 2022.

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

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.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 9786–9796. PMLR, 2020.

Binxu Wang and Carlos R Ponce. The geometry of deep generative image models and its applications. In *ICLR*, 2021.

Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *CVPR*, pp. 11379–11388, 2022.

Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402. Ieee, 2003.

Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, pp. 18430–18439, 2022.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 7354–7363. PMLR, 2019.

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

Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. In *ECCV*, 2022.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, pp. 597–613. Springer, 2016.

A Appendix

A.1 More details of datasets and settings

FFHQ The FFHQ (Karras et al., 2019) dataset contains 70,000 face images. It is only used for training the initial checkpoint for the generator ($\mathbf{G}$) and the encoders in our framework.

CelebA-HQ CelebA-HQ (Karras et al., 2018; Liu et al., 2015) contains 24,183 training face images and 2,824 testing images. For a fair comparison with previous inversion methods, we only use the test split 2,824 images for testing. In this paper, since all of the testing images from this dataset are from the real human face, we did not present the qualitative visualizations for privacy protection. We only present the quantitative comparisons in the paper.

StyleGAN2-Fake In order to present the rendering results qualitatively without using real faces, we use the fake yet very realistic faces released by (Karras et al., 2020b). This dataset contains 263 images of resolution 1,024×1,024 of very realistic human faces generated by StyleGAN2 (Karras et al., 2020b). We present the testing results qualitatively using these images.

AFHQ Besides the experiments of GAN inversion on human faces, we also conduct the experiments on animal faces using AFHQ (Choi et al., 2020) and present the results later in the appendix. This dataset contains 15,000 high-quality images at 512×512 resolution and includes three categories of animals which are cats, dogs, and wildlife. Each category has about 5000 images. For each category, the dataset split around 500 images as a test set and provide all remaining images as a training set.

A.2 Implementation Details

All training and testing images are resized to size $256 \times 256 \times 3$, denoting width, height, and channel respectively. The experimental style-based NeRF generator (G) employs the checkpoint of StyleNeRF (Gu et al., 2021) with dimension 256. The base encoder E_{base} employs a series of residual blocks and 1 linear projection layer. The residual encoder E_{res} employs the architecture from pSp (Richardson et al., 2021) and we set the number of residual iterations as 3 in the experiments. We set the dimension of the latent code w as 512 which is the same as the generator and the number of latent code of $w+$ as 17 following the checkpoint from StyleNeRF (Gu et al., 2021). For the hyperparameter for all of the loss functions, all of losses are equally weighted ($\lambda_{feat}^{base} = 1.0$, $\lambda_{img}^{base} = 1.0$, $\lambda_{feat}^{res} = 1.0$ and $\lambda_{img}^{res} = 1.0$) for all the experiments. The batch size is set as 32 where 16 is for synthesized images and 16 is for real images. In the 16 synthesized images in each batch, we sample 4 identity latent w_{syn} from StyleNeRF (Gu et al., 2021) and each w_{syn} samples 4 camera poses (randomly and uniformly sample yaw angle in the range $-35°$ to $+35°$ and roll angle as 0 for simplicity), which can be formulated into these 16 synthesized images. We optimize the network using Adam optimizer with the learning rate set as 0.0001. Each experiment is conducted on 1 Nvidia GPU A100 (80G) with a batch size of 32 and implemented in PyTorch. We now present more details about the model architecture below:

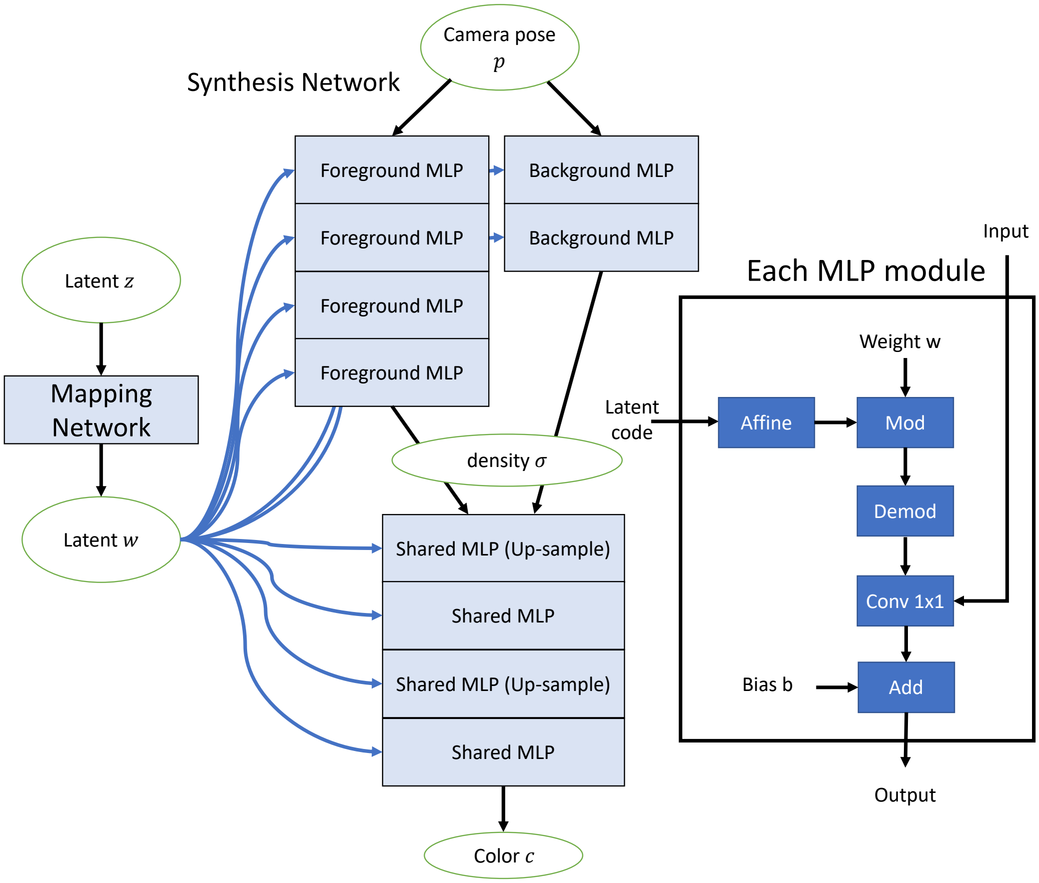

Figure 7: Brief overview of the architecture of StyleNeRF (Gu et al., 2021).

Generator (StyleNeRF) StyleNeRF (Gu et al., 2021) has a mapping network and a synthesis network as StyleGAN (Karras et al., 2019) does. The overview of the network is roughly presented in Figure 7. For the mapping network, latent codes are sampled from standard Gaussian distribution and processed by a number of fully connected layers. The synthesis network employs NeRF++ which consists of a unit sphere for foreground NeRF and a background NeRF using inverted sphere parameterization. Two MLPs that represent foreground and background are used to predict the density. The color prediction is performed using another shared MLP. Each style-conditioned MLP block consists of an affine transformation layer and a 1×1 convolution layer. The convolution weights are modulated with the affine-transformed styles and then demodulated for computation. Leaky-ReLU is used as non-linear activation. We directly utilize the checkpoint provided by StyleNeRF (Gu et al., 2021) without further change on the network and the pre-trained weights. More details can be found at (Gu et al., 2021).

Base Encoder As mentioned earlier, the base encoder E_{base} contains 6 residual blocks and 1 linear projection layer. The output of the encoder will be a vector of 512-dimension w latent code. The network is roughly presented in Figure 8. Since not all of the testing data in the real world has ground truth pose from the off-the-shelf model, our base encoder can also predict the yaw and roll angles from the input image while training with the ground-truth pose outputs from the synthesized images. The output dimension will be 514 (512 plus 2) if the additional task for pose prediction is added.

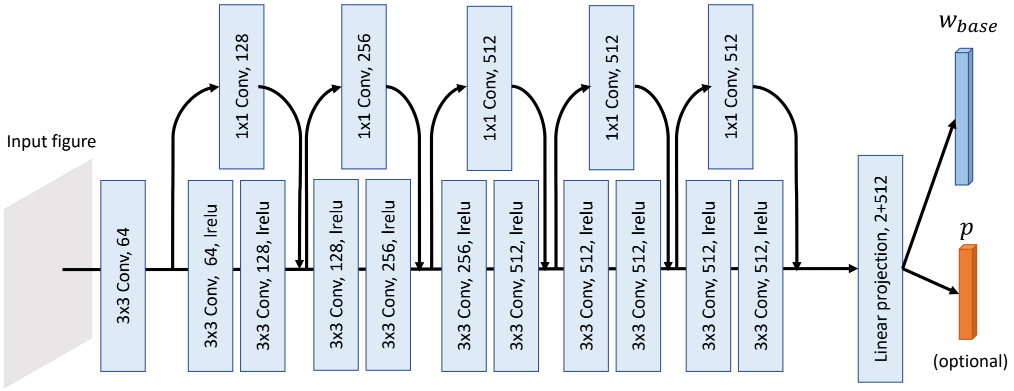

Figure 8: Brief overview of the architecture of base encoder.

Residual Encoder The overview of the network is roughly presented in Figure 9. The encoder derives the style input latent codes from three intermediate feature maps of spatial resolutions 16 × 16 (for input index 0 to 2), 32 × 32 (for input index 3 to 6), and 64 × 64 (for index 7 to last one). Each style vector is obtained from the corresponding feature map using a Map2style block, which is a convolutional network containing a series of 2-strided convolutions with LeakyReLU activations. More details can be found at (Richardson et al., 2021).

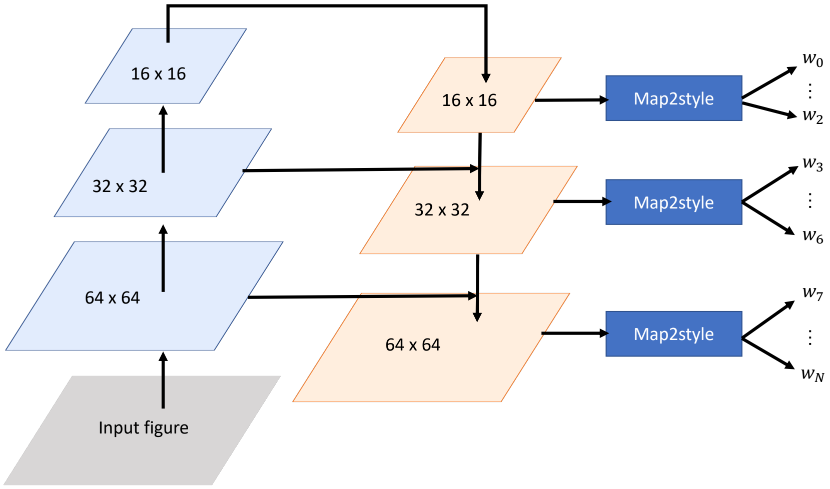

Figure 9: Brief overview of the architecture of residual encoder using pSp (Richardson et al., 2021).

A.3 Extension to support online optimization

In this section, we would like to analyze the effectiveness and the possibility of our encoders for supporting online optimization. We conduct the experiments of utilizing our model for producing initial style latent code in both latent vector optimization to $\mathcal{W}+$ (Karras et al., 2020b) and PTI (Roich et al., 2021). The results and comparison are presented in Figure 10. We can observe that, for both of the optimization approaches, our encoders improve the identity and 3D preservation for both image reconstruction and novel-view rendering (-35° in the examples).

A.4 More ablation studies and results

The outputs of the base encoder and the residual encoder. To further analyze the importance of both our base encoder and the residual encoder, we also visualize the output of these two encoders. The visualization is presented in Figure 11. The output of $\hat{x}_{base}$ can be seen as using the encoding into $\mathcal{W}$ in the fourth column of Figure 5 plus the feature-level loss. Though the generated $\hat{x}_{base}$ has a

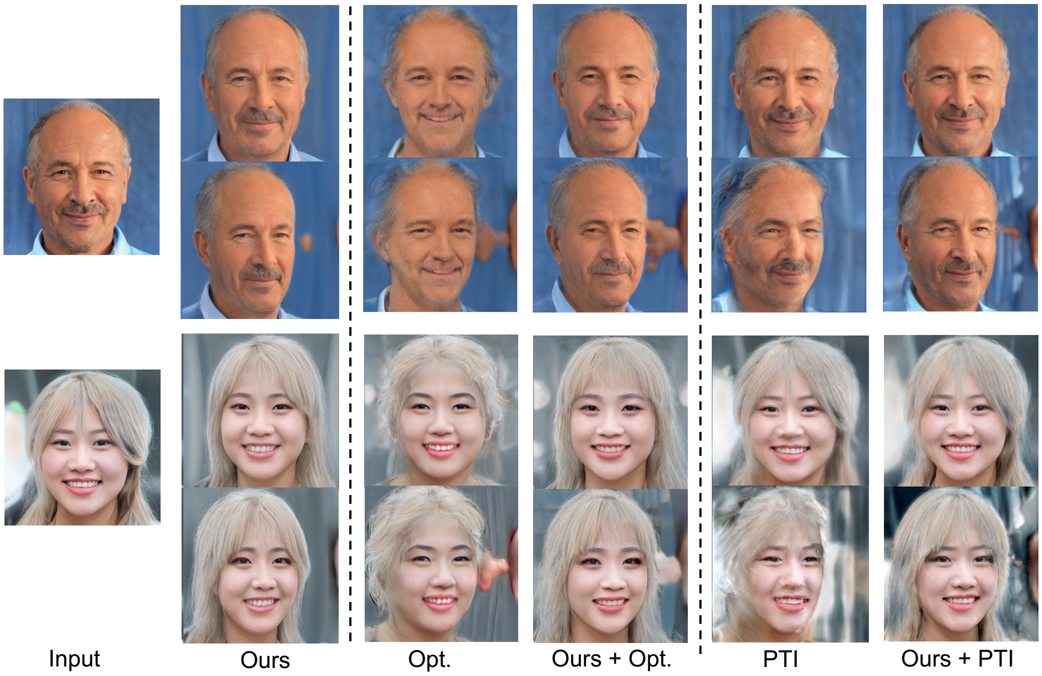

Figure 10: **The effectiveness of our model for online optimization.** We utilize the starting latent produced by our model for Opt. (optimization to $\mathcal{W}+$ (Karras et al., 2020b)) and PTI (Roich et al., 2021).

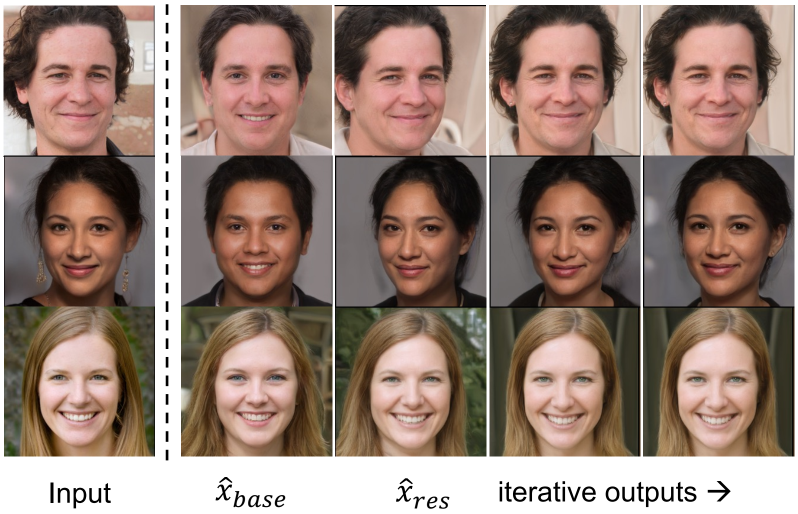

Figure 11: **Ablation studies on the outputs of each stage.** The testing images are from StyleGAN-Fake.

gap from the input image, it preserves 3D view consistency. Then we can produce the latent code for generating $\hat{x}_{res}$ on top of $\hat{x}_{base}$ with more fine details. We also demonstrate that our restyle steam can also be done in several iterations yet does not improve the latent code as much as Restyle (Alaluf et al., 2021) presented in 2D StyleGAN.

More qualitative results on novel rendering. We present more results on novel view rendering for different input images in Figure 12. This figure demonstrates the generalization of our encoders plus the StyleNeRF generator to different races, gender, age, and skin tones.

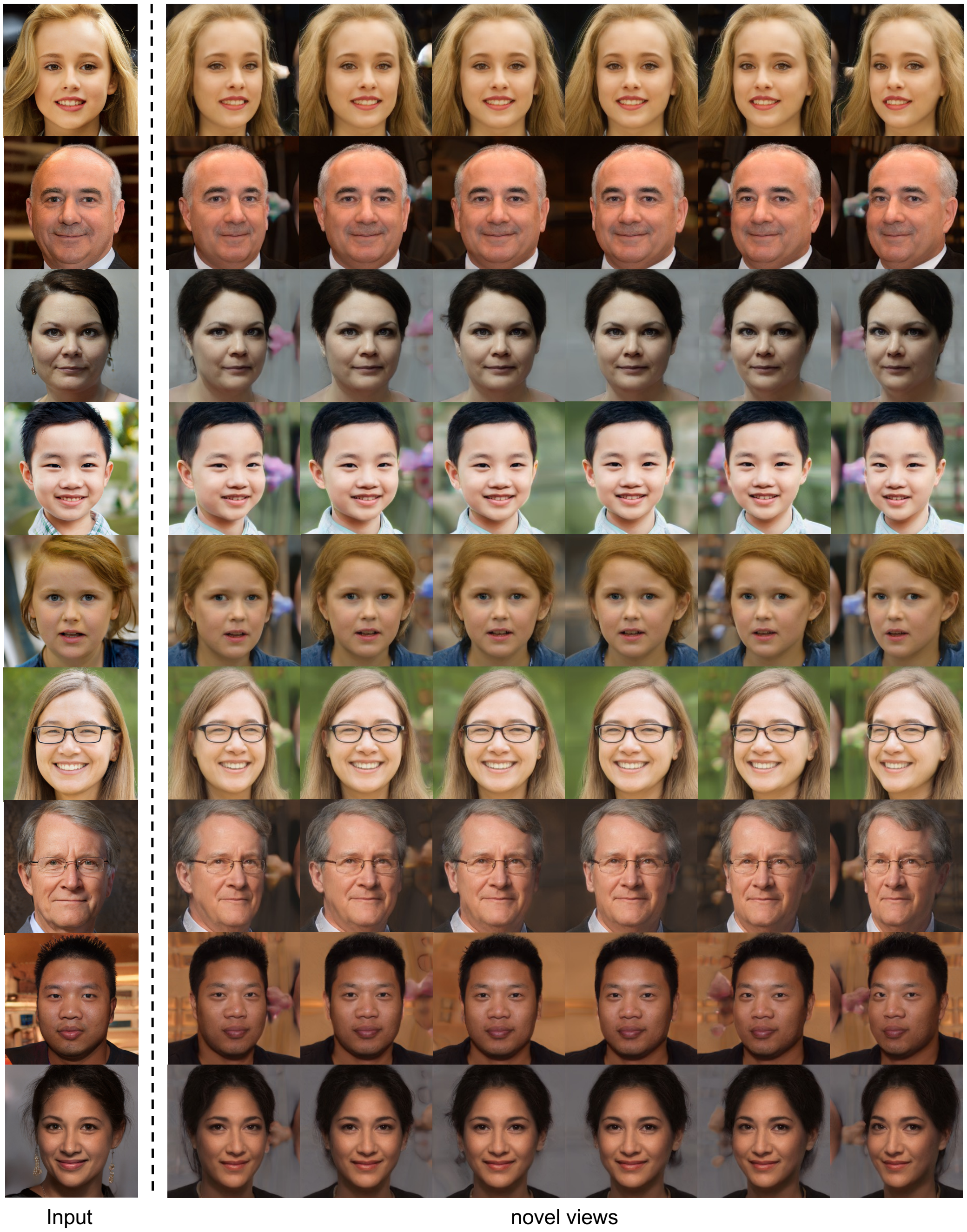

Figure 12: More qualitative results using our encoder for novel view rendering on StyleGAN2-Fake.

Generator: StyleNeRF (Gu et al., 2021) vs. EG3D (Chan et al., 2022) To analyze the significance of the generators for the GAN inversion, we also compare the results replacing the StyleNeRF generator with Eg3D using the same input image from Figure 12. EG3D (Chan et al., 2022) is composed of StyleGAN2 architecture and utilizes tri-plane volume rendering. More details can be referred to in their paper. We re-train the encoders using the generator EG3D (Chan et al., 2022) and present the results of novel-view rendering in Figure 14.

A.5 More experiments with AFHQ and settings

To analyze the ability of our model on the inversion of animal faces, we conduct the experiments using AFHQ (Choi et al., 2020). This dataset includes three categories of animals which are cats, dogs, and wildlife. Since StyleNeRF (Gu et al., 2021) does not release the checkpoint for this dataset, we train our own checkpoint using the open-source code ourselves which may have sub-optimal rendering effectiveness. In addition, since we do not have a suitable off-the-shelf pose estimator for the animals, we additionally train the pose encoder in our base encoder (as shown in Figure 8) for estimating the camera pose for the animals. We present the results of GAN inversion using our encoders in Figure 13. In addition, we also present the GAN inversion using the checkpoint of cat (a subset of AFHQ) released by EG3D (Chan et al., 2022). These two figures show that our framework is able to perform effective 3D-aware GAN inversion on animal faces.

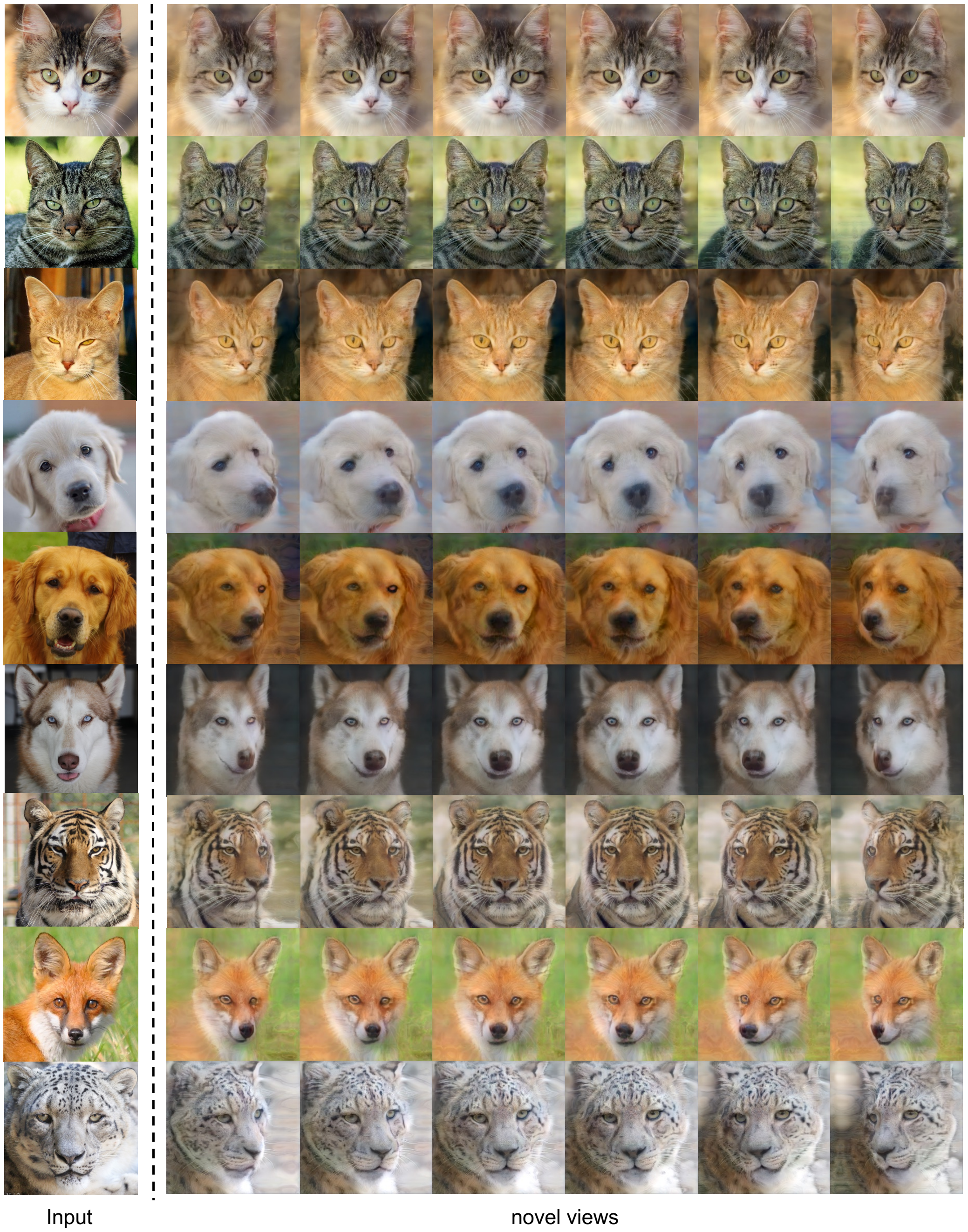

Figure 13: Additinal qualitative results using our encoder for novel view rendering on AFHQ. Note that since the checkpoint of StyleNeRF for AFHQ is not released, we train a sup-optimal checkpoint ourselves.

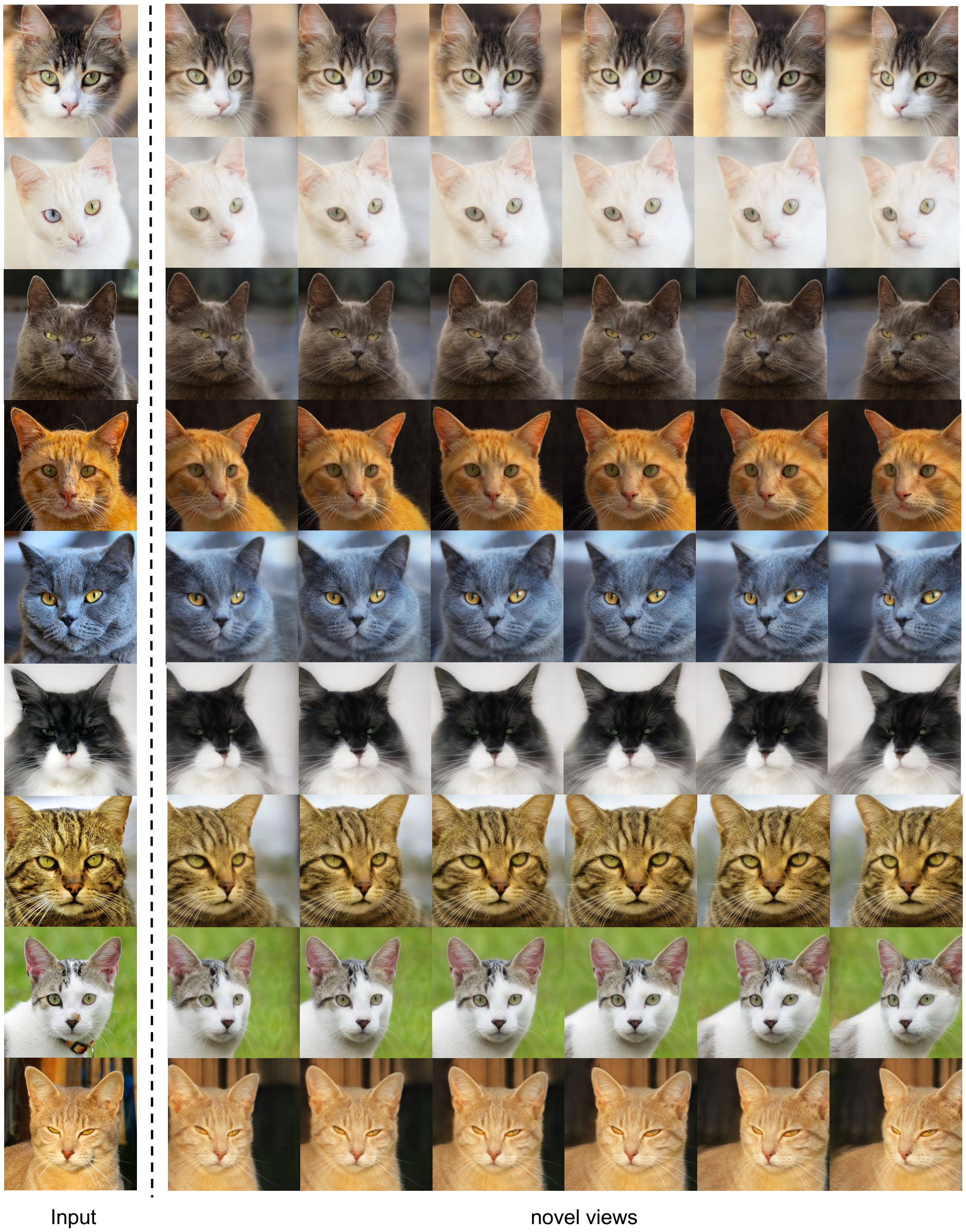

Figure 14: Additinal qualitative results using our encoder for novel view rendering on Cats subset in AFHQ replacing StyleNeRF with **Eg3D** as the generator (**G**).