3D GAN Inversion with Pose Optimization

Jaehoon Ko* Kyusun Cho* Daewon Choi Kwangrok Ryoo Seungryong Kim†

Korea University, South Korea

{kjh9604, kyustorm7, daeone0920, kwangrok21, seungryong_kim}@korea.ac.kr

Abstract

With the recent advances in NeRF-based 3D aware GANs quality, projecting an image into the latent space of these 3D-aware GANs has a natural advantage over 2D GAN inversion: not only does it allow multi-view consistent editing of the projected image, but it also enables 3D reconstruction and novel view synthesis when given only a single image. However, the explicit viewpoint control acts as a main hindrance in the 3D GAN inversion process, as both camera pose and latent code have to be optimized simultaneously to reconstruct the given image. Most works that explore the latent space of the 3D-aware GANs rely on ground-truth camera viewpoint or deformable 3D model, thus limiting their applicability. In this work, we introduce a generalizable 3D GAN inversion method that infers camera viewpoint and latent code simultaneously to enable multi-view consistent semantic image editing. The key to our approach is to leverage pre-trained estimators for better initialization and utilize the pixel-wise depth calculated from NeRF parameters to better reconstruct the given image. We conduct extensive experiments on image reconstruction and editing both quantitatively and qualitatively, and further compare our results with 2D GAN-based editing to demonstrate the advantages of utilizing the latent space of 3D GANs. Additional results and visualizations are available at `https://3dgan-inversion.github.io/`.

1. Introduction

Recent Generative Adversarial Network (GAN) [14] architectures show incredible results in synthesizing unconditional images with a diverse range of attributes. Especially, StyleGAN [23, 24] has achieved photorealistic visual quality on high-resolution images. Moreover, several works have explored the latent space $\mathcal{W}$ and found its disentangled properties, which enable the control of certain image features and semantic attributes such as gender or hair color. However, its real-world application is only possible with GAN inversion by bridging the generated image space with real image domain. GAN inversion inverts a real image back into the latent space of a pre-trained GAN, extending the manipulation capability of the model to real images.

However, editing an image by projecting it onto the latent space of a 2D GAN makes the task vulnerable to the same set of problems of 2D GANs. As training methods of 2D GANs do not take into account the underlying geometry of an object, they offer limited control over the geometrical aspects of the generated image. Thus, manipulating image viewpoint using the latent space of 2D GANs always runs into the issue of multi-view inconsistency.

On the other hand, 3D-aware image synthesis addresses this issue by integrating explicit 3D representations into the generator architecture and enabling explicit control over the camera pose. With the success of neural radiance fields (NeRF) [31] in novel view synthesis, recent 3D-aware generation models employ a NeRF-based generator, which condition the neural representations on sample noise or disentangled appearance and shape codes in order to represent diverse 3D scenes or object. More recent attempts address the quality gap between 3D GANs and 2D GANs by adopting 2D CNN-based upsampler or efficient point sampling strategy, which enables the generation of high-resolution and photorealistic images on par with 2D GANs.

Projecting a 2D image onto the learned manifold of these 3D GANs unlocks many opportunities in computer vision applications. Not only can it generate multi-view consistent images from the acquired latent code, but it can also gather the exact surface geometry from the image. Furthermore, recent 3D GANs adopted a style-based generator module to learn the disentangled representations of 3D geometry and appearance. Similar to the latent-based image editing tasks of 2D GANs, by manipulating the latent code of a style-based 3D-aware generator, we can manipulate the semantic attributes of the reconstructed 3D model.

Despite its usefulness, few research has been conducted

*Equal contribution.
†Corresponding Author.

on 3D GAN inversion. Since 3D-aware GANs initially require a random vector and camera pose for their image generation, an inversion process to reacquire the latent code of a given image necessitates the camera pose of the image, information that real-life images usually often lack. Most of the existing methods require ground-truth camera information or must rely on the off-the-shelf geometry and camera pose from the 3D morphable model, which limits their application to a single category.

In this work, we propose a 3D-GAN inversion method that iteratively optimizes both the latent code and 3D camera pose of a given image simultaneously. We build upon the recently proposed 2D GAN inversion method that first inverts the given image into a pivot code, and then slightly tunes the generator based on the fixed pivot code (i.e. *pivotal tuning* [38]), which showed prominent results in both reconstruction and editability. Similarly, we acquire both the latent code and camera pose simultaneously as a pivot and fine-tune the pre-trained 3D-GAN to alter the generator manifold into pivots. Note that this is non-trivial since shape and camera direction compromise each other during optimization.

Recognizing the interdependency between latent code and camera parameter, we use a hybrid of learning and optimization-based approach by first using an encoder to infer a rough estimate of the camera pose and latent code, and further refining it to an optimal destination. As can be seen in our experiments, giving a good initial point for optimizing pivots much less falls into the local minimum. In order to further enforce the proximity of the camera viewpoint, we introduce regularization loss that utilizes traditional depth-based image warping [51].

We demonstrate that our method enables high-quality reconstruction and editing while preserving multi-view consistency, and show that our results are applicable to a multitude of different categories. While we evaluate our proposed method on EG3D [7], the current state-of-the-art 3D-aware GAN, our method is also relevant to other 3D-aware GANs that leverage NeRF for its 3D representation.

2. Related Work

Generative 3D-Aware Image Synthesis. 3D-aware GANs aim to generate 3D-aware images from 2D image collections. The first approaches utilize voxel-based representation [32], which lacks fine details in image generation, due to memory inefficiency from its 3D representation. Starting from [39], several works achieved better quality by adopting NeRF-based representation, even though they struggle on generating high-resolution images due to the expensive computational cost of volumetric rendering. Some approaches proposed an efficient point sampling strategy [39, 11, 46], while others adopted 2D CNN-layers to efficiently upsample the volume rendered feature map [34, 15, 50]. Recently, other methods proposed hybrid representations to reduce the computational burden from MLP layers, while achieving high-resolution image generation [7, 47, 40, 42]. Especially, our work is implemented on EG3D [7], which achieved state-of-the-art image quality while preserving 3D consistency.

2D GAN Inversion. The first step in applying latent-based image editing on real-world images is to project the image to the latent space of pre-trained GANs. Existing 2D GAN inversion approaches can be categorized into optimization-based, learning-based, and hybrid methods. Optimization approaches [1, 9] directly optimize the latent code for a single image. This method can achieve high reconstruction quality but is slow for inference. Unlike per-image optimization, learning-based approaches [37, 43, 3] use a learned encoder to project images. These methods have shorter inference time but fail to achieve high-fidelity reconstruction. Hybrid approaches are a proper mixture of the two aforementioned methods. [16, 53] used the cooperative learning strategy for encoder and direct optimization. PTI [38] fine-tunes StyleGAN parameters for each image after obtaining an initial latent code, solving the trade-off [43] between reconstruction and editability.

3D GANs Inversion. 3D GAN inversion approaches share the same goal as 2D GAN inversion with the additional need for extrinsic camera parameters. Few existing methods solving inverse problems in 3D GANs propose effective training solutions of their own. [10] proposed regularization loss term to avoid generating unrealistic geometries by leveraging the popular CLIP [36] model. [27] can animate the single source image to resemble the target video frames, but is limited to human face as it requires off-the-shelf models [13] to extract expression, pose, and shape. [6] proposes a joint distillation strategy for training encoder, which is inadequate for 3D GANs that contain mapping function.

Image Manipulation. Image manipulation can be conducted by changing the latent code derived from GAN inversion. Many works have examined semantic direction in the latent spaces of pre-trained GANs and then utilized it for editing. While some works [41, 2] use supervision in the form of semantic labels predicted by off-the-shelf attribute classifiers or annotated images, they are often limited to known attributes. Thus, other researchers resorted to using an unsupervised approach [17] or contrastive learning based methods [48, 35] to find meaningful directions. In this work, we leverage GANSpace [17], which performs principal component analysis in the latent space, to demonstrate latent-based manipulation of 3D shape.

3. Preliminaries

2D GANs Inversion and Pivotal Tuning. Given a pre-trained 2D generator $\mathcal{G}_{\text{2D}}(\cdot\,;\theta)$ parameterized by weights θ, 2D GANs inversion aims to find the latent representation $\mathbf{w}$ that can be passed to the generator to reconstruct a given image x:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathcal{L}(x, \mathcal{G}_{\text{2D}}(\mathbf{w};\theta)), \tag{1}$$

where the loss function $\mathcal{L}(\cdot,\cdot)$ is usually defined as pixel-wise reconstruction loss or perceptual loss [49] between the given image x and reconstructed image $\mathcal{G}_{\text{2D}}(\mathbf{w};\theta)$.

To improve the performance, some other methods aim to optimize an encoder $\mathcal{E}(x;\theta_{\mathcal{E}})$ with parameters $\theta_{\mathcal{E}}$ that maps images to their latent representations such that:

$$\theta^*_{\mathcal{E}} = \underset{\theta_{\mathcal{E}}}{\operatorname{argmin}}\ \mathcal{L}(x, \mathcal{G}_{\text{2D}}(\mathcal{E}(x;\theta_{\mathcal{E}});\theta)). \tag{2}$$

Some recent methods [53, 5, 52] take a hybrid approach of leveraging the encoded latent representation $\theta_{\mathcal{E}}(x;\theta_{\mathcal{E}})$ with learned parameters $\theta_{\mathcal{E}}$ as an initialization for a subsequent optimization process for (1), resulting in a faster and more accurate reconstruction.

Furthermore, it has been recently well studied that existing GANs inversion methods [43, 54, 2] struggle on the trade-offs between reconstruction and editability. To overcome this, [38] proposed a pivotal tuning stage in a manner that after finding the optimal latent representation $\mathbf{w}^*$, called the pivot code, the generator weights θ are fine-tuned so that the pivot code can more accurately reconstruct the given image while keeping its editability:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}(x, \mathcal{G}_{\text{2D}}(\mathbf{w}^*;\theta)). \tag{3}$$

By utilizing the pivot code $\mathbf{w}^*$ and the tuned weights θ^*, the final reconstruction is obtained as $y^* = \mathcal{G}_{\text{2D}}(\mathbf{w}^*;\theta^*)$.

NeRF and 3D-aware GANs. Neural Radiance Fields (NeRF) [31] achieves a novel view synthesis by employing a fully connected network to represent implicit radiance fields that maps location and direction $(\mathbf{x}, \mathbf{d})$ to color and density $(\mathbf{c}, \sigma)$. Specifically, along with each projected ray r for a given pixel, M points are sampled as $\{t_i\}_{i=1}^{M}$, and with the estimated color and density $(\mathbf{c}_i, \sigma_i)$ of each sampled point, the RGB value $c(r)$ for each ray can be calculated by volumetric rendering as:

$$c(r) = \sum_{i=1}^{M} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i, \tag{4}$$

where $T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$, and δ_i is the distance between adjacent sampled points such that $\delta_i = t_{i+1} - t_i$. Furthermore, per-ray depth $d(r)$ can also be approximated as

$$d(r) = \sum_{i=1}^{M} T_i(1 - \exp(-\sigma_i\delta_i))t_i. \tag{5}$$

While NeRF trains a single MLP on multiple posed images of a single scene, NeRF-based generative models [15, 7] often condition the MLP on latent style code $\mathbf{w}$ that represents individual latent features learned from unposed image collections. These style-based 3D GANs have been popularly used in 3D aware image generation [15, 7, 50], and we denote this generator $\mathcal{G}_{\text{3D}}(\mathbf{w}, \pi;\theta)$ with a given latent code $\mathbf{w}$, which can be formally represented as the conditional function: $\{c, d\} = \mathcal{G}_{\text{3D}}(\mathbf{w}, \pi;\theta)$, where c is rendered RGB image and d is depth map.

4. Method

4.1. Overview

Our objective, which we call 3D GAN inversion, is to project a real photo into the learned manifold of the GAN model. However, finding the exact match for $\mathbf{w}^*$ and π^* for a given image x is a non-trivial task since one struggles to optimize if the other is drastically inaccurate. To overcome this, we follow [53, 5, 52] to first construct two encoders that approximately estimate the initial codes from x by $\mathbf{w} = \mathcal{E}(x;\theta_{\mathcal{E}})$ and $\pi = \mathcal{P}(x;\theta_{\mathcal{P}})$ (Sec. 4.2), and further solving optimization problem(Sec. 4.3). In particular, we introduce loss functions employed in (Sec. 4.3) and further discuss the effects and purposes of employing these loss functions(Sec. 4.4). An overview of our method is shown in Fig. 1.

4.2. Latent Encoder $\mathcal{E}$ and Pose Estimator $\mathcal{P}$

For better 3D GANs inversion, utilizing a well-trained estimator for initialization should be considered [19, 44], but it is a solution limited to a single category. To obtain category-agnostic estimator, we first generate a pseudo dataset and its annotation pair $\{(\mathbf{w}_{\text{ps}}, \pi_{\text{ps}}), x_{\text{ps}}\}$ to pre-train our encoders, where $x_{\text{ps}} = \mathcal{G}_{\text{3D}}(\mathbf{w}_{\text{ps}}, \pi_{\text{ps}};\theta)$. Thanks to the generation power of 3D-aware GANs, we can generate nearly unlimited numbers of pairs within the generator's manifold. More specifically, for given latent encoder $\mathcal{E}$, let $\triangle\mathbf{w} = \mathcal{E}(x_{\text{ps}};\theta_{\mathcal{E}})$ denote the output of the encoder, where $\mathbf{w} \in \mathbb{R}^{1\times 512}$. Following the training strategy of [43], we employ the generator $\mathcal{G}_{\text{3D}}$ and its paired discriminator $\mathcal{D}$ to guide encoder to find the best replication of x_{ps} with $\bar{\mathbf{w}} + \triangle\mathbf{w}$, where $\bar{\mathbf{w}}$ is an average embeddings of $\mathcal{G}_{\text{3D}}$. We provide more detailed implementation procedure of pre-training each network in the Appendix.

4.3. Optimization

After the pre-training step, given an image x, the learnable latent vector and camera pose are first initialized

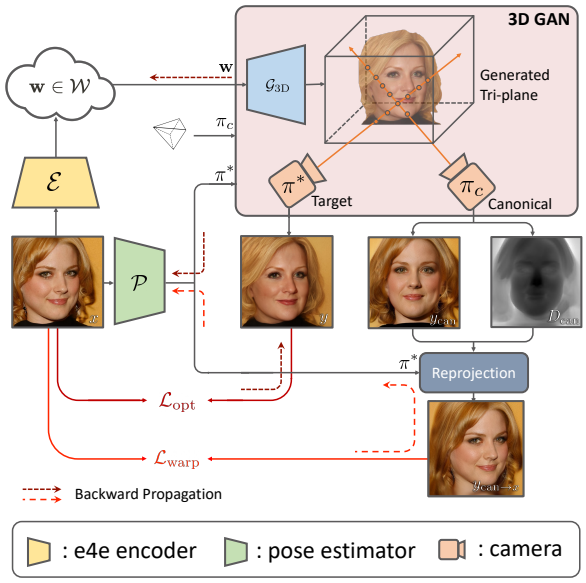

Figure 1: **Overall architecture.** This figure shows our method using depth-based warping to optimize latent code and camera pose simultaneously 4.3.

from trained estimators as $\mathbf{w}_{\text{init}} = \bar{\mathbf{w}} + \mathcal{E}(x;\theta_{\mathcal{E}})$ and $\pi_{\text{init}} = \mathcal{P}(x;\theta_{\mathcal{P}})$. Subsequently, they are further refined for a more accurate reconstruction. In this stage, we reformulate optimization step in (1) into the 3D GAN inversion task, in order to optimize the latent code and camera viewpoint starting from each initialization $\{\mathbf{w}_{\text{init}}, \pi_{\text{init}}\}$, such that:

$$\mathbf{w}^*, \pi^*, n^* = \underset{\mathbf{w},\pi,n}{\text{argmin}}\ \mathcal{L}^{\text{opt}}(x, \mathcal{G}_{3D}(\mathbf{w}, \pi, n; \theta)), \tag{6}$$

where n denotes the per-layer noise inputs of the generator and $\mathcal{L}^{\text{opt}}$ contains employed loss functions on the optimization step. Note that, following the latent code optimization method in [38], we use the native latent space $\mathcal{W}$ which provides the best editability.

In addition, in the pivotal-tuning step, using the optimized latent code $\mathbf{w}^*$ and optimized camera pose π^*, we augment the generator's manifold to include the image by slightly tuning $\mathcal{G}_{3D}$ with following reformulation of (3):

$$\theta^* = \underset{\theta}{\text{argmin}}\ \mathcal{L}^{\text{pt}}(x, \mathcal{G}_{3D}(\mathbf{w}^*, \pi^*, n^*; \theta)). \tag{7}$$

In this optimization, following [38], we unfreeze the generator and tune it to reconstruct the input image x with given $\mathbf{w}^*$ and π^*, which are both constant. We also implement the same locality regularization in [38]. Again, $\mathcal{L}^{\text{pt}}$ denotes a combination of loss functions on the pivotal-tuning step.

4.4. Loss functions

LPIPS and MSE Loss. To reconstruct the given image x, we adopt commonly used LPIPS loss for both optimization and pivotal tuning step. As stated in (6), the loss is used to train the both latent code $\mathbf{w}$ and camera pose π. Additional mean square error is given only at the pivotal tuning step, which is commonly used to regularize the sensitivity of LPIPS to adversarial examples. Formally, our losses can be defined by:

$$\mathcal{L}_{\text{lpips}} = \mathcal{L}_{\text{lpips}}(x, \mathcal{G}^{\text{c}}_{3D}(\mathbf{w}, \pi, n; \theta)), \tag{8}$$

$$\mathcal{L}_{L2} = \mathcal{L}_{L2}(x, \mathcal{G}^{\text{c}}_{3D}(\mathbf{w}, \pi, n; \theta)). \tag{9}$$

Depth-based Warping Loss. Similar to [51], every point on the target image can be warped into other viewpoints. We consider the shape representation of $\mathbf{w}$ latent code plausible enough to fit in the target image. Given a canonical viewpoint π_{can}, we generate a pair of image and depth map $\{y_{\text{can}}, D_{\text{can}}\} = \mathcal{G}_{3D}(\mathbf{w}, \pi_{\text{can}}; \theta)$. Let $y_{\text{can}}(r)$ denote the homogeneous coordinates of a pixel γ in the generated image of a canonical view π_{can} using (4). Also for each $y_{\text{can}}(r)$ we obtain the depth value $D_{\text{can}}(r)$ by using (5).

Then we can obtain $y_{\text{can}}(r)$'s projected coordinates onto the source view π_x denoted as $\hat{y}_x(r)$ by

$$\hat{y}_x(r) \sim K \hat{\pi}_{\text{can}\to x} D_{\text{can}}(r) K^{-1} y_{\text{can}}(r), \tag{10}$$

where K is the camera intrinsic matrix, and $\hat{\pi}_{\text{can}\to x}$ is the predicted relative camera pose from canonical to source. As $\hat{x}(r)$ for every pixel r are continuous values, following [51], we exploit the differentiable bilinear sampling mechanism proposed in [21] to obtain the projected 2D coordinates.

For simplicity of notation, from a generated image $y_{\text{can}} = \mathcal{G}^{\text{c}}_{3D}(\mathbf{w}, \pi_{\text{can}}; \theta)$, we denote the projected image as $y_{\text{can}\to x} = y_{\text{can}}\langle proj(D_{\text{can}}, \pi_{\text{can}\to x}, K)\rangle$, where $proj(\cdot)$ is the resulting 2D image using the depth map D_{can} and $\langle\cdot\rangle$ denotes the bilinear sampling operator, and define the objective function to calculate $\pi_{\text{can}\to x}$:

$$\mathcal{L}_{\text{warp}} = \mathcal{L}_{\text{lpips}}\left(x, y_{\text{can}}\langle proj(D_{\text{can}}, \pi, K)\rangle\right), \tag{11}$$

again using an LPIPS loss to compare the two images.

Depth Regularization Loss. Neural radiance field is infamous for its poor performance when only one input view is available. Although tuning the parameters of 2D GANs seem to retain its latent editing capabilities, we found the NeRF parameters to be much more delicate, and tuning them to a single view degrades the 3D structure before reaching the desired expressiveness, resulting in low-quality renderings at novel views.

To mitigate this problem, we take advantage of the geometry regularization used in [33] and encourage the generated depth to be smooth, even from unobserved viewpoints. The regularization is based on the real-world

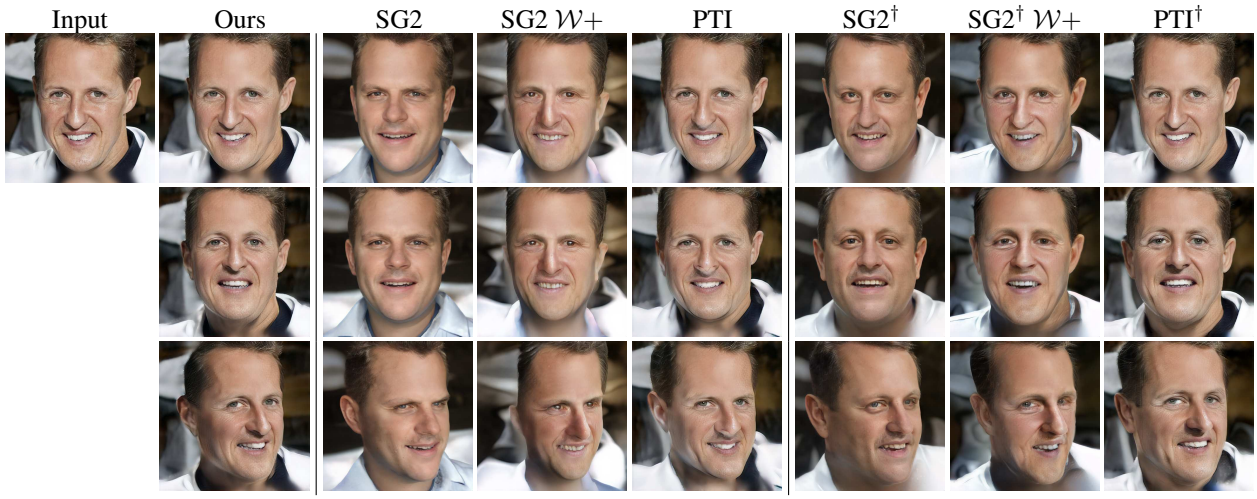

Figure 2: **Comparison of novel view synthesis of out-of-domain samples.** Given the optimized camera pose $\hat{\pi}$ and latent code $\hat{\mathbf{w}}$ obtained by each method, we explicitly control the viewpoint of the generated facial scene, by differing π for different camera viewpoint. We compare our 3D GAN inversion method to standard 2D GAN inversion methods by applying the gradient-based optimization to both the latent code and the camera pose. We also leverage the same methods only on the latent code with the given ground-truth camera pose, and show the results labeled with †.

observation that real geometry or depth tends to be smooth, and is more likely to be flat, and formulated such that depth for each pixel should not be too different from those of neighboring pixels. We enforce the smoothness of the generated depth $D(r)$ for each pixel r with the depth regularization loss:

$$\mathcal{L}_{DR}(D) = \sum_{i,j=1}^{H-1,W-1} \Big(\big(D(r_{i,j}) - D(r_{i+1,j})\big)^2 + \big(\big(D(r_{i,j}) - D(r_{i,j+1})\big)^2 \Big), \tag{12}$$

where H and W are the height and width of the generated depth map, and $r_{i,j}$ indicates the ray through pixel (i, j). Note that while [33] implements the geometry regularization by comparing overlapping patches, we utilize the full generated depth map D for our implementation.

Overall Loss Function. Ultimately, we define the entire optimization step with a generated image and depth $\{y, D\} = \mathcal{G}_{\text{3D}}(\mathbf{w}, \pi, n; \theta)$:

$$\begin{aligned} \mathcal{L}^{\text{opt}} = \; & \mathcal{L}_{\text{lpips}}(x, y) \\ & + \lambda_{\text{warp}} \mathcal{L}_{\text{warp}}(x, y_{\text{can}}, D) + \lambda_n \mathcal{L}_n(n), \end{aligned} \tag{13}$$

and the pivotal tuning process is defined by:

$$\begin{aligned} \mathcal{L}^{\text{pt}} = \; & \mathcal{L}_{\text{lpips}}(x, y) \\ & + \lambda_{L2} \mathcal{L}_{L2}(x, y) + \lambda_{DR} \mathcal{L}_{DR}(D), \end{aligned} \tag{14}$$

where $\mathcal{L}_n$ denotes the noise regularization loss proposed in [25], which prevents the noise from containing crucial signals of the target image.

5. Experimental Results

5.1. Experimental Settings

Datasets. We conduct the experiments on two 3D object types, *human faces* and *cat faces*, as they are the two most popular tasks in GAN inversion. For all experiments, we employ the pre-trained EG3D [7] generator. For *human faces*, we use the weights pre-trained on the cropped FFHQ dataset [23], and we evaluate our method with the CelebA-HQ validation dataset [22, 29]. We also use the pre-trained weights on the AFHQ dataset [8] for *cat faces* and evaluate on the AnimalFace10 dataset [28].

Baselines. Since the current works [10, 27] do not provide public source code for reproduction and comparison of their work, we mainly compare our methods with the popular 2D GAN inversion methods: The direct optimization scheme proposed by [25] to invert real images to $\mathcal{W}$ denoted as SG2, a similar method but extended to $\mathcal{W}+$ space [1] denoted as SG2 $\mathcal{W}+$, and the PTI method from [38]. We adopt these methods to work with the pose-requiring 3D-aware GANs, either by providing the ground-truth camera pose during optimization or using the same gradient descent optimization method for the camera pose.

5.2. Reconstruction

Quantitative Evaluation. For quantitative evaluation, we reconstruct 2,000 validation images of CelebA-HQ and utilize the same standard metrics used in 2D GAN inversion literature: pixelwise L2 distance using MSE,

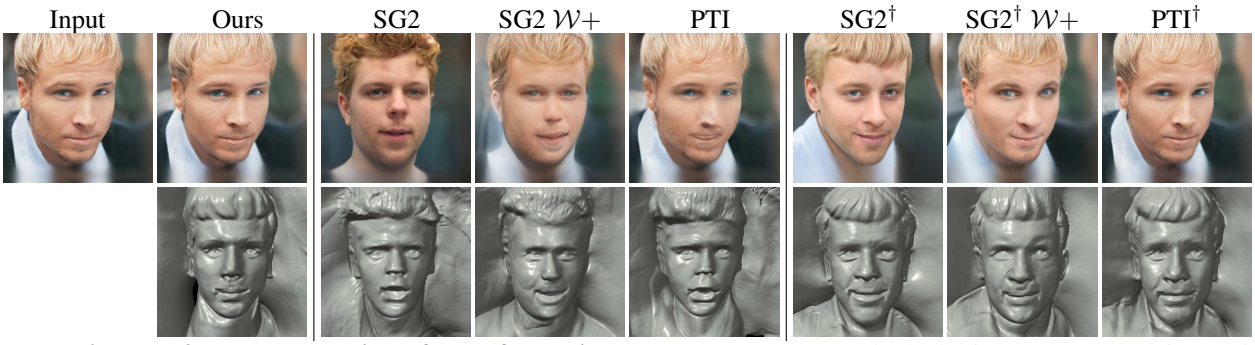

Figure 3: **2D and 3D Reconstruction of out-of-domain samples.** We compare both the image reconstruction and 3D reconstruction capabilities of each method, where the 3D shapes are iso-surfaces extracted from the density field using marching cubes. Methods labeled with † use ground-truth camera pose.

Method	MSE↓	LPIPS↓	MS-SSIM↑	ID Sim.↑	FID ↓
SG2	0.0277	0.3109	0.5889	0.0957	36.0291
SG2 $\mathcal{W}+$	0.0163	0.2398	0.6833	0.2906	32.3971
PTI	0.0036	0.0789	0.8221	0.6671	32.7366
SG2†	0.0232	0.2898	0.6151	0.1125	34.7612
SG2† $\mathcal{W}+$	0.0117	0.2029	0.7349	0.3972	31.1732
PTI†	**0.0033**	**0.0722**	**0.8309**	0.6737	**28.5911**
Ours	0.0035	0.0777	0.8280	**0.7013**	30.1192

Table 1: **Qualitative reconstruction results** measured over the CelebA-HQ test set. The **best** and runner-up values are marked bold and underlined, respectively. Methods labeled with † use ground-truth camera pose.

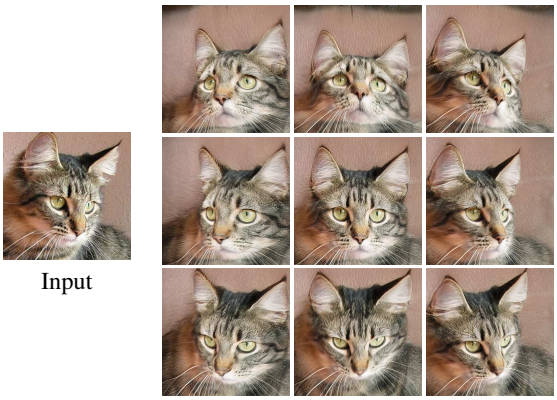

Figure 4: **Reconstruction and novel view synthesis on AnimalFace10 dataset.** Our method is not limited to human face and can be applied to other domains.

perceptual similarity metric using LPIPS [49] and structural similarity metric using MS-SSIM [45]. In addition, for facial reconstruction, we follow recent 2D GAN inversion works [12, 4] and measure identity similarity using a pre-trained facial recognition network of CurricularFace [20]. Furthermore, we measure the 3D reconstruction ability of our method, as the clear advantage of 3D GAN inversion over 2D GAN inversion is that the former allows for novel view synthesis given a single input image. In other words, the latent code acquired by a successful inversion process should be able to produce realistic and plausible images at random views. In order to measure image quality, we calculate the Frechet Inception Distance (FID) [18] between the original images and 2,000 generated images from randomly sampled viewpoints.

The results are shown in Table 1. As can be seen, compared to the 2D GAN inversion methods that use the same gradient-based optimization for camera pose, using the depth-based warping method better guides the camera viewpoint to the desired angle, showing higher reconstruction metrics. Furthermore, while methods designed for high expressiveness such as SG2 $\mathcal{W}+$ and PTI achieve comparable pixel-wise reconstruction abilities, our method has the upper hand when it comes to 3D reconstruction, producing better quality images for novel views of the same face. Even compared to inversion methods using ground-truth camera pose, our method achieves competitive results without external data and reliably predicts the camera pose, showing similar reconstruction scores for each metric.

Qualitative Evaluation. We visualize the reconstruction and novel view synthesis results in Fig. 2. While our method performs significantly better at generating images in novel views, our method also achieves comparable results even to methods using ground-truth camera pose. Not only do we provide qualitative comparison of the visual quality of inverted images, but we also show the reconstructed 3D shape of the given image as a mesh using the Marching Cubes algorithm [30], as demonstrated in Fig. 2. Different from 2D GAN inversion, we also compare the 3D geometry quality of each face by comparing the rendered views of different camera poses. Furthermore, we evaluate reconstruction and novel view synthesis on *cat faces* in Fig. 4

While SG2 $\mathcal{W}+$ and PTI show reasonable reconstruction results at the same viewpoint, when we steer the 3D model

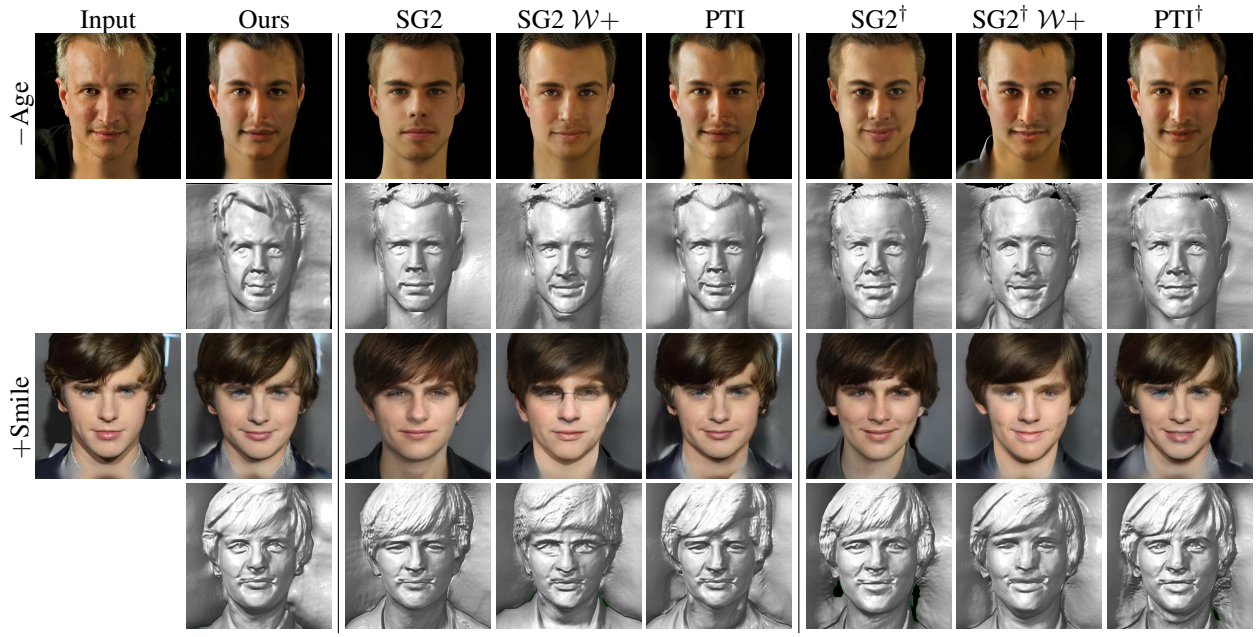

Figure 5: **Editing Quality Comparison.** We perform various edits [17] over latent codes and camera pose acquired by each method. Benefiting from the capabilities of 3D-aware GANs, we also compare edited 3D shape generated from the edited latent codes. Our method achieves both realistic and accurate manipulation and is also more capable of preserving the identity and geometry of the original input. Methods labeled with † use ground-truth camera pose.

to different viewpoints, the renderings are incomplete and show visual artifacts with degraded 3D consistency. In contrast, we can see that by using our method, we can synthesize novel views with comparable quality to methods requiring ground-truth viewpoints.

5.3. Editing Quality

We employ GANSpace [17] method to quantitatively evaluate the manipulation capability of the acquired latent code. In Fig. 5, we compare latent-based editing results to the 2D GAN-inversion methods used directly to 3D-aware GANs. Consistent with 2D GANs, the latent code found in the $\mathcal{W}+$ space produces more accurate reconstruction but fails to perform significant edits, while latent codes in $\mathcal{W}$ space show subpar reconstruction capability. Using pivotal tuning [38] preserves the identity while maintaining the manipulation ability of $\mathcal{W}$ space. Reinforcing [38] with our more reliable pose estimation and regularized geometry, our method best preserves the 3D-aware identity while successfully performing meaningful edits. We also provide quantitative evaluation in the supplementary material.

5.4. Comparison with 2D GANs

We demonstrate the effectiveness and significance of 3D-aware GAN inversion by comparing the viewpoint manipulation using the explicit camera pose control for EG3D and latent space. Even though [26] points out that 3D GANs lack the ability to manipulate semantic attributes, recent advancements of NeRF-based generative architectures have achieved a similar level of expressiveness and editability to 2D GANs. As recent 3D-aware GANs such as EG3D can generate high-fidelity images with controllable semantic attributes while maintaining a reliable geometry, image editing using 3D GANs offers consistent multi-view results which are more useful.

Figure 6: **Simultaneous attribute editing and viewpoint shift comparison of 2D and 3D GANs.** We compare editing results of applying attribute editing (smile) and viewpoint interpolation at the same time on the latent code acquired by PTI [38] on StyleGAN2 [25] and the latent code acquired by our method on EG3D [7].

In Fig. 6, we compare the simultaneous manipulation abilities of 2D and 3D-aware GANs. While pose manipulation of 2D GANs only allows for implicit control by the editing magnitude, 3D-aware GANs enable explicit

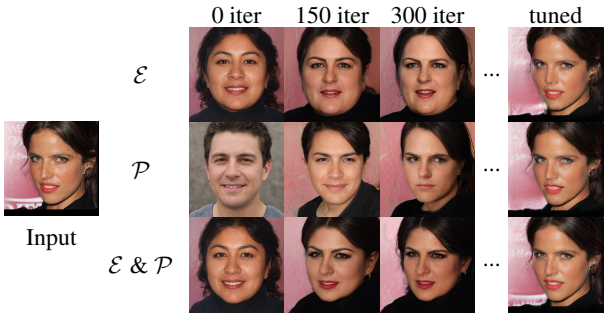

Figure 7: **Importance of initialization.** We selectively employ the latent encoder $\mathcal{E}$ and pose estimator $\mathcal{P}$ and compare their optimization process.

$\mathcal{E}$	$\mathcal{P}$	MSE↓	LPIPS↓	MS-SSIM↑	ID Sim.↑	θ	ϕ
✗	✓	0.0036	0.0782	0.8263	0.6958	3.19	2.90
✓	✗	0.0038	0.0790	0.8219	0.6810	5.73	5.93
✓	✓	0.0035	0.0777	0.8280	0.7013	3.16	2.70

Table 2: **Importance of initialization.** We state the importance of utilizing pre-trained networks as an initialization for optimization, by comparing the optimization results that started from network outputs(✓) and those that started from random initialization(✗).

control of the viewpoint. Also, the edited images in 2D GANs are not view-consistent and large editing factors result in undesired transformations of the identity and editing quality. In contrast, pose manipulation of 3D GANs is always multi-view consistent, thus producing consistent pose interpolation even for an edited scene.

5.5. Ablation Study

Importance of initialization. We test the effectiveness of our design by comparing our full hybrid method to the single-encoder methods and show the results in Fig. 7 and Table 2. We show that using a hybrid approach consisting of a learned encoder $\mathcal{E}$ and gradient-based optimization is the ideal approach when obtaining the latent code. Similarly, leveraging a pose estimator $\mathcal{P}$ for initialization for the pose refinement also shortens the optimization time.

Effectiveness of Geometry Regularization. We study the role of depth smoothness regularization in the pivotal tuning step by varying the weight λ_{DR}. We show the generated geometry and its pixelwise MSE value after fine-tuning the generator in Fig. 8. While solely using the reconstruction loss leads to better quantitative results, novel views still contain floating artifacts and the generated geometry has holes and cracks. In contrast, including the depth smoothness regularization with its weight as $\lambda_{DR} = 1$ enforces solid and smooth surfaces while producing

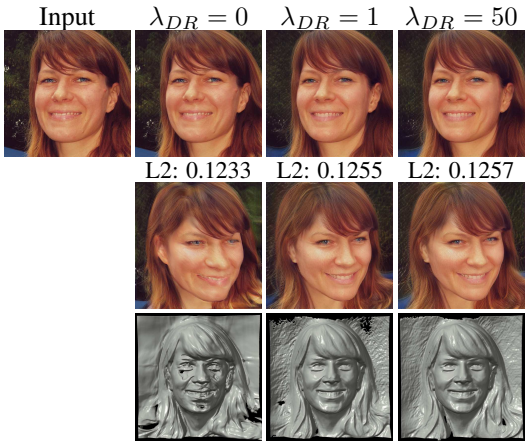

Figure 8: **Comparison of different weights of depth smoothness loss during the pivotal tuning stage.** Although excluding depth smoothness shows the best reconstruction result (first row), the reconstructed geometry is distorted (third row) and provides malformed renderings for novel views (second row). Best viewed in zoom.

accurate scene geometry. It should be noted that a high weight for the depth smoothness blurs the fine segments of the generated geometry such as hair.

6. Conclusion

We present a geometric approach for inferring both the latent representation and camera pose of 3D GANs for a single given image. Our method can be widely applied to the 3D-aware generator, by utilizing hybrid optimization method with additional encoders trained by manually built pseudo datasets. In essence, this pre-training session helps the encoder acquire the representation power and geometry awareness of 3D GANs, thus finding a stable optimization pathway. Moreover, we utilize several loss functions and demonstrate significantly improved results in reconstruction and image fidelity both quantitatively and qualitatively. It should be noted that whereas previous methods used 2D GANs for editing, our work suggests the possibility of employing 3D GANs as an editing tool. We hope that our approach will encourage further research on 3D GAN inversion, which will be further utilized with the single view 3D reconstruction and semantic attribute editing.

Acknowledgements. This research was supported by the MSIT, Korea (No.2021-0-00155, Context and Activity Analysis-based Solution for Safe Childcare, No.2021-0-02068, Artificial Intelligence Innovation Hub), and National Research Foundation of Korea (NRF-2021R1C1C1006897).

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019.

[2] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 2021.

[3] 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.

[4] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *CVPR*, 2022.

[5] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *ICCV*, 2019.

[6] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *CVPR*, 2022.

[7] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

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

[9] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[10] Giannis Daras, Wen-Sheng Chu, Abhishek Kumar, Dmitry Lagun, and Alexandros G Dimakis. Solving inverse problems with nerfgans. *arXiv preprint arXiv:2112.09061*, 2021.

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

[12] Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *CVPR*, 2022.

[13] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *SIGGRAPH*, 2021.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[15] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2022.

[16] Shanyan Guan, Ying Tai, Bingbing Ni, Feida Zhu, Feiyue Huang, and Xiaokang Yang. Collaborative learning for faster stylegan embedding. *arXiv preprint arXiv:2007.01758*, 2020.

[17] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[19] Heng-Wei Hsu, Tung-Yu Wu, Sheng Wan, Wing Hung Wong, and Chen-Yi Lee. Quatnet: Quaternion-based head pose estimation with multiregression loss. *IEEE Transactions on Multimedia*, 2019.

[20] 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*, 2020.

[21] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015.

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

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

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

[26] Jeong-gi Kwak, Yuanming Li, Dongsik Yoon, Donghyeon Kim, David Han, and Hanseok Ko. Injecting 3d perception of controllable nerf-gan into stylegan for editable portrait image synthesis. *arXiv preprint arXiv:2207.10257*, 2022.

[27] 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.

[28] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *ICCV*, 2019.

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[30] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 1987.

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

[32] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019.

[33] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022.

[34] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021.

[35] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021.

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

[37] 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*, 2021.

[38] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021.

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

[40] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *arXiv preprint arXiv:2206.07695*, 2022.

[41] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

[42] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022.

[43] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM*, 2021.

[44] Roberto Valle, José M Buenaposada, and Luis Baumela. Multi-task head pose estimation in-the-wild. *TPAMI*, 2020.

[45] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *ACSSC*. Ieee, 2003.

[46] Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds. *arXiv preprint arXiv:2206.07255*, 2022.

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

[48] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *ICCV*, 2021.

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

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

[51] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

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

[53] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016.

[54] Peihao Zhu, Rameen Abdal, Yipeng Qin, John Femiani, and Peter Wonka. Improved stylegan embedding: Where are the good latents?, 2020.