TriPlaneNet: An Encoder for EG3D Inversion

Ananta R. Bhattarai Matthias Nießner Artem Sevastopolsky

Technical University of Munich (TUM)

Abstract

Recent progress in NeRF-based GANs has introduced a number of approaches for high-resolution and high-fidelity generative modeling of human heads with a possibility for novel view rendering. At the same time, one must solve an inverse problem to be able to re-render or modify an existing image or video. Despite the success of universal optimization-based methods for 2D GAN inversion, those applied to 3D GANs may fail to extrapolate the result onto the novel view, whereas optimization-based 3D GAN inversion methods are time-consuming and can require at least several minutes per image. Fast encoder-based techniques, such as those developed for StyleGAN, may also be less appealing due to the lack of identity preservation. Our work introduces a fast technique that bridges the gap between the two approaches by directly utilizing the tri-plane representation presented for the EG3D generative model. In particular, we build upon a feed-forward convolutional encoder for the latent code and extend it with a fully-convolutional predictor of tri-plane numerical offsets. The renderings are similar in quality to the ones produced by optimization-based techniques and outperform the ones by encoder-based methods. As we empirically prove, this is a consequence of directly operating in the tri-plane space, not in the GAN parameter space, while making use of an encoder-based trainable approach. Finally, we demonstrate significantly more correct embedding of a face image in 3D than for all the baselines, further strengthened by a probably symmetric prior *enabled during training.*

Figure 1. For a given picture, our method predicts the appropriate latent code and the tri-plane offsets for the EG3D generator in a feed-forward manner. This way, both the frontal view and the novel view rendering can be obtained with high fidelity and in close to real time.

1. Introduction

In recent years, numerous works [7, 8] have tackled the problem of multi-view consistent image synthesis with 3D-aware GANs. Such methods make generators aware of a 3D structure by modeling it with explicit voxel grids [15,30,40] or neural implicit representations [8, 32]. Most notably, EG3D [7] introduced a 3D GAN framework based on a tri-plane 3D representation that is both efficient and expressive to enable high-resolution 3D-aware image synthesis. Moreover, they demonstrate state-of-the-art results for unconditional geometry-aware image synthesis.

The main applications of 3D GANs include human face inversion, including head tracking, reenactment, facial manipulation, and novel view synthesis of a given image or video. Oftentimes, the classical GAN formulation does not support trivial inversion, i.e. finding the appropriate code in the learned GAN space for a given sample. A straightforward way to achieve this is by obtaining the latent code of the input image via optimization-based or encoder-based approaches, i.e. applying 2D GAN inversion techniques. An existing branch of research studies 2D GAN inversion

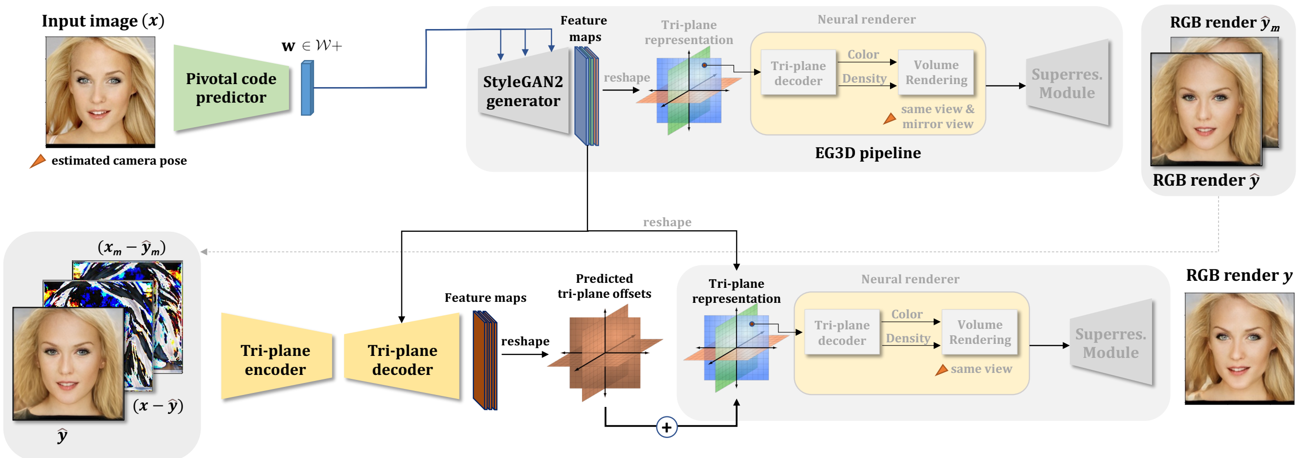

Figure 2. Overview of the proposed method. TriPlaneNet consists of two branches. The first branch (above) comprises the predictor $\hat{w} = \boldsymbol{\phi}(x)$ of the pivotal latent code $\hat{w} \in \mathcal{W}+$, which results in an input-view RGB image $\hat{y} = \mathcal{R}(\boldsymbol{G}(\hat{w}), \pi)$ and a mirror-view RGB image $\hat{y}_m = \mathcal{R}(\boldsymbol{G}(\hat{w}), \pi_m)$ with corresponding camera matrices π and π_m respectively after passing it through the EG3D tri-plane generator $\boldsymbol{G}(\cdot)$ and renderer block $\mathcal{R}(\cdot, \cdot)$ containing super-resolution module. The second branch (below) uses the first-stage approximation $\hat{y}$ its difference with the target $(x - \hat{y})$, difference between mirror image and rendered mirror view image $(x_m - \hat{y}_m)$ and the tri-planes features $\boldsymbol{G}(\hat{w})$ to predict the numerical offsets to the tri-planes $\Delta \boldsymbol{T}$ by a convolutional autoencoder $\Delta \boldsymbol{T} = \boldsymbol{\psi}(\hat{y}, x - \hat{y})$, which yields the final prediction $y = \mathcal{R}(\boldsymbol{G}(\hat{w}) + \Delta \boldsymbol{T}, \pi)$.

in high detail [2, 3, 5, 36, 41, 53], but nevertheless, the problem remains underexplored in 3D.

Optimization-based inversion methods are often superior to encoder-based approaches in terms of reconstruction quality. However, encoder-based techniques are orders of magnitude faster as they map a given image to the latent space of GAN in a single forward pass. Compared to 2D GAN inversion, 3D GAN inversion is a more challenging task as the inversion needs to both preserve the identity of an input image and plausibly embed the head in 3D space. In particular, optimization-based 2D GAN inversion methods that have no knowledge of the specific GAN architecture make sure to yield a high-quality rendering of the desired image from the same camera view, but the lack of any geometry information in the image may produce broken or stretched geometry when rendered from a novel camera. Optimization-based 3D GAN inversion techniques improve these shortcomings by adding 3D constraints in the optimization process. Even though these techniques prevent geometry collapse and offer high-fidelity reconstruction, they are slow and time-consuming. We improve the above-mentioned shortcomings in two separate ways. First, by predicting an input latent code for the EG3D generator with a convolutional encoder, we observe that the geometry is preserved better than by optimizing it. This can be attributed to the fact that the encoder, trained for the inversion task, is exposed to thousands of images under different poses and, in this way, learns to be 3D-aware. Second, we utilize the knowledge about the model and improve the details and consistency by predicting offsets to the tri-planes that constitute the 3D representation in EG3D. Unlike voxel grids or implicit representations, tri-planes can be naturally estimated by 2D convnets and, as demonstrated by our experiments, can realistically express object features beyond the capabilities of an input latent code, e.g., hands and long hair (see Fig. 1). This advantage is attained by recovering the object representation directly in the world space. Since the tri-plane offsets are fully predicted by convolutional layers, our inversion can run in close to real time on modern GPUs.

We propose the EG3D-specific inversion scheme in two stages. In the first stage, the initial inversion is obtained using the latent encoder that directly embeds the input image into the $\mathcal{W}+$ space of EG3D. In the second stage, we introduce another encoder, TriPlaneNet, that learns to refine the initial reconstruction. Conditioned on the input image and corresponding tri-plane features, it predicts a numerical offset for them. The system is trained with a combination of perceptual and photometric losses. In addition, we make use of the soft constraint based on the mirror image – *probably symmetric prior* inspired by [46] – that makes the encoder even more 3D-aware.

To summarize, our contributions are the following:

- We propose a novel and fast inversion framework for EG3D that enables high-quality reconstruction and plausible geometric embedding of a head in 3D space by directly utilizing the tri-plane representation and a soft symmetry constraint.

- We demonstrate that our method achieves on-par reconstruction quality compared to optimization-based inversion methods and is an order of magnitude time faster. Our method is also more resilient towards harder cases, such as when a hat or accessories are featured.

2. Related Work

3D Generative Models for Human Faces. Representing and generating diverse 3D human faces and heads attracted increasing attention over the last decade [9, 16, 29], while the appearance of NeRF [27] has sparked additional interest in that topic. The first generative models built upon NeRF-style volumetric integration [31,38] achieved generalization by conditioning the multi-layer perceptron on latent codes, representing the object's shape and appearance. Later introduced π-GAN [8] and StyleNeRF [13] condition the generative network on the output of the StyleGAN-like generator [20], which amounted to the higher-quality rendering of faces and arbitrary objects with subtle details. As a next major improvement step, authors of EG3D [7] propose a tri-plane 3D representation that serves as a bridge between expressive implicit representations and spatially-restricting explicit representations. As a byproduct, methods such as EG3D and StyleSDF [32] allow for the extraction of explicit, highly detailed geometry of the human faces, despite the fact that they are trained without any volumetric supervision. Further, recently demonstrated abilities of diffusion models to generate highly accurate 2D images are currently being transferred onto 3D objects [28, 51] and 3D human heads [33, 44].

GAN Inversion. Unlike other kinds of generative models, such as VAE or normalizing flows, inverting a GAN (finding the appropriate latent code for a given image) is oftentimes a tricky and computationally demanding task. Early attempts focused on the tuning of the latent code with the optimization-based approaches [10, 20, 24]. Various approaches exploited the idea of predicting latent representation by an encoder [14, 26, 34, 35, 54]. In [37], a universal PTI method is introduced, which comprises the optimization of a latent code and, consequently, fine-tuning parameters of the generator. A recent survey on GAN inversion [47] compares multiple generic techniques introduced since the appearance of GANs.

Inversion of 2D GAN. For StyleGAN, an important observation was made by the authors of [2] that operating in the extended $\mathcal{W}+$ space is significantly more expressive than in the restrictive $\mathcal{W}$ generator input space. The latter idea has been strengthened and better adapted for face editing with the appearance of pSp [36] and e4e [42], as well as of their cascaded variant ReStyle [5] and other works [3, 41, 53]. Similarly to PTI but in an encoder-based setting, HyperStyle [6] and HyperInverter [12] predict offsets to the StyleGAN generator weights in a lightweight manner in order to represent the target picture in a broader space of parameters.

Inversion of 3D GAN. Unlike the 2D case, the inversion of a 3D GAN is a significantly more advanced problem due to the arising ambiguity: the latent code must be both compliant with the target image and correspond to its plausible 3D representation. While PTI remains a universal method that solves this problem for an arbitrary generator, recent art demonstrates that the quality rapidly declines when the PTI inversion result is rendered from a novel view. The suggested ways of resolving this fidelity-consistency trade-off for an arbitrary 3D GAN include incorporating multi-view consistency or geometry regularizers [23, 48], augmenting training with surrogate mirrored images [49], introducing local features [22], or optimizing camera parameters and latent code simultaneously [21]. All of these approaches are still optimization-based and require at least a few minutes of inference time per image. A concurrent encoder-based work Live 3D Portrait [43] also leverages the tri-plane representation for high-fidelity reconstruction while relying on a self-constructed generator instead of EG3D and skipping the latent code prediction part. Our work solves the inversion for the pre-trained, frozen EG3D generator and addresses face manipulation due to the use of the latent space. Additionally, in [43], the training pipeline is reversed compared to ours. Starting from the random latent code, they generate synthetic images from EG3D to train the encoder. In contrast, we pass real and synthetic images through the encoder to generate latent codes. Another work, EG3D-GOAE [50], concurrent to ours, modifies the internal features of EG3D instead of tri-planes directly.

3. Method

3.1. Preliminaries

GAN inversion. Given a target image x, the goal of GAN inversion is to find a latent code that minimizes the reconstruction loss between the synthesized image and the target image:

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(x, G(w;\theta)) \tag{1}$$

where $G(w;\theta)$ is the image generated by a pre-trained generator G parameterized by weights θ, over the latent w. The problem in (1) can be solved via optimization or encoder-based approaches. Encoder-based approaches utilize an encoder network E to map real images into a latent code. The training of an encoder network is performed over a large set of images $\{x^i\}_{i=1}^N$ to minimize:

$$\min_E \sum_{i=1}^{N} \mathcal{L}(x^i, G(E(x^i);\theta)) \tag{2}$$

During inference, an input image is inverted by $G(E(x);\theta)$. In the recent works [6, 12, 37], a number of approaches are

Figure 3. Qualitative comparison for image reconstruction. Compared to other approaches, our method can reconstruct a face in the same view in more detail, especially introducing more detail for features such as hats, hair, and background.

proposed to additionally estimate image-specific generator parameters $\theta(x)$ by a convolutional network.

EG3D. EG3D [7] uses tri-plane 3D representation for geometry-aware image synthesis from 2D images. EG3D image generation pipeline consists of several modules: a StyleGAN2-based feature generator, a tri-plane representation, a lightweight neural decoder, a volume renderer, and a super-resolution module. To synthesize an image, a random latent code $z \in \mathbb{R}^D$ (typically, $D = 512$) and camera parameters are first mapped to a pivotal latent code $w \in \mathcal{W}+$ using a mapping network. Then, w is fed into the StyleGAN2 CNN generator $\boldsymbol{G}(\cdot)$ to generate a $H \times W \times 96$ feature map. This feature map is reshaped to form three 32-channel planes, thus forming a tri-plane feature representation $\boldsymbol{T}$ of the corresponding object. To sample from the tri-plane features, a position $p \in \mathbb{R}^3$ is first projected onto the three feature planes. Then, corresponding feature vectors $(F_{xy}(p), F_{xz}(p), F_{yz}(p))$ are retrieved using bilinear interpolation and aggregated. These aggregated features are processed by a lightweight neural decoder to transform the feature into the estimated color and density at the location p. Volume rendering is then performed to project 3D feature volume into a feature image. Finally, a super-resolution module is utilized to upsample the feature image to the final image size. For simplicity, we will later refer to the lightweight neural decoder, renderer, and the super-resolution block, all combined, as the rendering block $\mathcal{R}(\cdot,\cdot)$. The high efficiency and expressiveness of EG3D, as well as the ability to work with tri-planes directly, motivates the development of our model-specific inversion algorithm.

pSp. Richardson *et al*. [36] proposed a pSp framework based on an encoder that can directly map real images into $\mathcal{W}+$ latent space of StyleGAN. In pSp, an encoder backbone with a feature pyramid generates three levels of feature maps. The extracted feature maps are processed by a map2style network to extract styles. The styles are then fed into the generator network to synthesize an image $\hat{y}$:

$$\hat{y} = G(E(x) + \bar{w}), \tag{3}$$

where $G(\cdot)$ and $E(\cdot)$ denote the generator and encoder net-

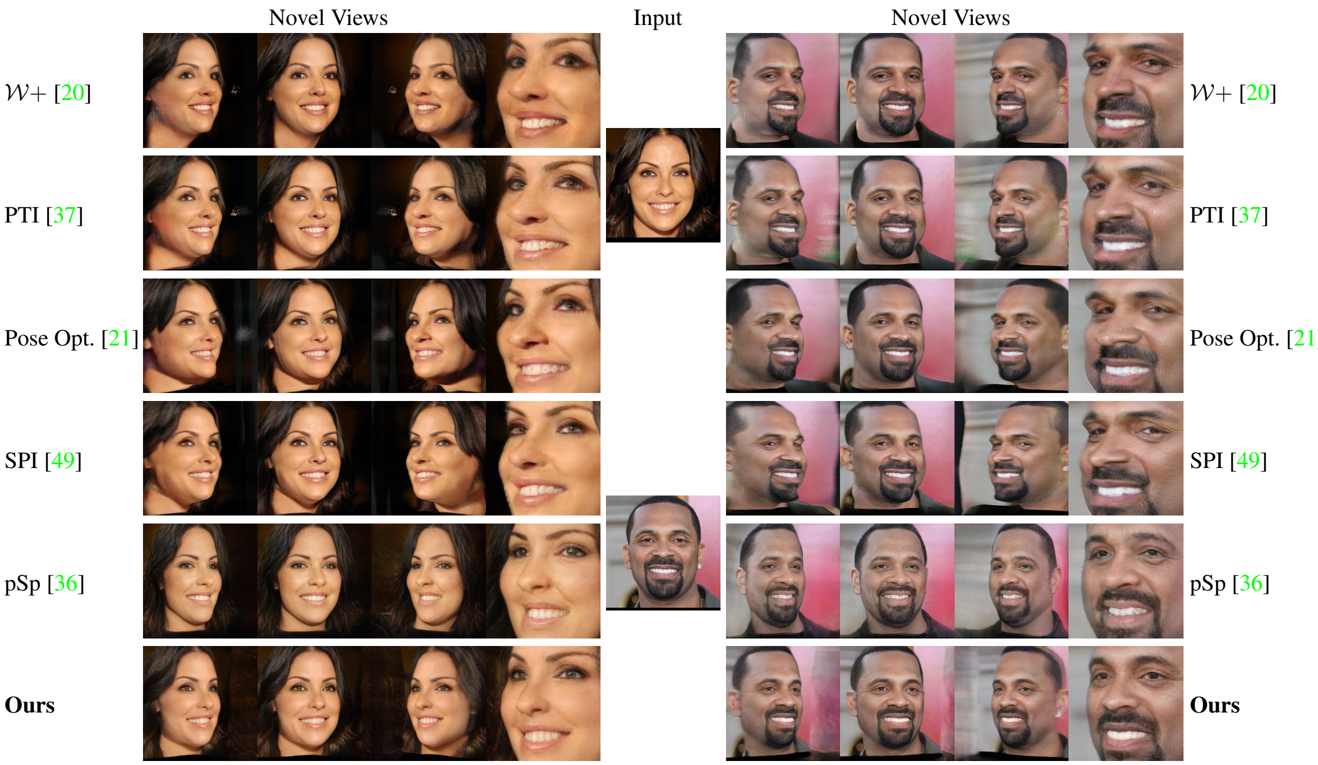

Figure 4. Qualitative evaluation on novel view rendering of yaw angle -0.6, -0.3, and 0.6 radians (full and zoom-in). In comparison to others, our method preserves identity and multi-view consistency better when rendered from a novel view.

works respectively and $\bar{w}$ is the average style vector of the pre-trained generator.

3.2. TriPlaneNet

Our TriPlaneNet inversion framework comprises two branches (see Fig. 2 for the overview). The first branch employs a latent encoder following a design of pSp to embed an input image into $\mathcal{W}+$ space of EG3D. Specifically, given an input image x, we train a encoder ϕ to predict the pivotal latent $\hat{w} \in \mathcal{W}+$:

$$\hat{w} = \phi(x) + \bar{w} \tag{4}$$

where the dimension of $\hat{w}$ is $K \times D$ (for the output image resolution of 128, $K = 14$, and $D = 512$). The pivotal code is then fed into StyleGAN2 generator $\boldsymbol{G}(\cdot)$ in the EG3D pipeline to obtain initial tri-plane features $\boldsymbol{T}$. Then, the tri-plane representation is processed by the rendering block $\mathcal{R}(\cdot,\cdot)$ to generate initial reconstruction $\hat{y}$:

$$\hat{y} = \mathcal{R}(\boldsymbol{G}(\hat{w}), \pi) \tag{5}$$

where π is the input-view camera matrix.

The second branch consists of a convolutional autoencoder ψ that learns to predict numerical offsets to the initial tri-plane features. The input to the encoder module of the autoencoder network is the channel-wise concatenation of initial reconstruction $\hat{y}$, the difference between an input image and initial input-view reconstruction $(x - \hat{y})$, and the difference between a mirrored input image and initial mirror-view reconstruction $(x_m - \hat{y}_m)$. The decoder takes input from the encoder and first branch tri-plane features. Given these inputs, the autoencoder is tasked with computing tri-plane offsets $\Delta\boldsymbol{T}$ with respect to tri-plane features obtained in the first branch:

$$\Delta\boldsymbol{T} = \boldsymbol{\psi}(\hat{y},\, x - \hat{y},\, x_m - \hat{y}_m,\, G(\hat{w})) \tag{6}$$

The new tri-plane features corresponding to the inversion of the input image x are then computed as an element-wise addition of tri-plane offsets $\Delta\boldsymbol{T}$ with initial tri-plane features $\boldsymbol{T} = G(\hat{w})$. This new tri-plane representation is similarly processed by the rendering block $\mathcal{R}(\cdot,\cdot)$ to obtain the final reconstructed image:

$$y = \mathcal{R}(\boldsymbol{T} + \Delta\boldsymbol{T}, \pi) \tag{7}$$

A detailed view of the architecture is presented in Supp. [1].

3.3. Loss Functions

The pipeline is trained by minimizing the loss function that decomposes into the separate loss expressions for two branches:

$$\mathcal{L}_{\phi,\psi}(x, y, \hat{y}, y_m, \hat{y}_m) = \mathcal{L}_{\phi}(x, \hat{y}, \hat{y}_m) + \mathcal{L}_{\psi}(x, y, y_m) \tag{8}$$

For training the encoder $\phi(\cdot)$ in the first branch, we employ pixel-wise $\mathcal{L}_2$ loss, LPIPS loss [52], and ID loss [11]. Therefore, the total loss formulation is given by

$$\begin{aligned}\mathcal{L}_{\phi}(x, \hat{y}, \hat{y}_m) = \lambda_1 \mathcal{L}_2(x, \hat{y}) + \lambda_2 \mathcal{L}_{\text{LPIPS}}(x, \hat{y}) \\ + \lambda_3 \mathcal{L}_{\text{id}}(x, \hat{y}) + \lambda_m \mathcal{L}_m(x_m, \hat{y}_m)\end{aligned} \tag{9}$$

where $x_m = \text{flip}(x)$, and $\mathcal{L}_m(x_m, \hat{y}_m)$ is a *probably symmetric prior* defined as

$$\begin{aligned}\mathcal{L}_m(x_m, \hat{y}_m) = \lambda_4 \mathcal{L}_{\text{symm}}(x_m, \hat{y}_m, \sigma(x_m)) \\ + \lambda_5 \mathcal{L}_{\text{LPIPS}}(x_m, \hat{y}_m) + \lambda_6 \mathcal{L}_{\text{id}}(x_m, \hat{y}_m),\end{aligned} \tag{10}$$

where $\mathcal{L}_{\text{symm}}$ is a symmetric term inspired by [46]. Since human faces are not perfectly symmetric, the symmetric term is based on a per-pixel Gaussian density with the pixel-wise uncertainty map $\sigma(x_m)$ that assigns lower confidence to the region in the mirrored image where the symmetry assumption fails.

The loss for the second branch $\mathcal{L}_{\psi}$ is constructed the same way as $\mathcal{L}_{\phi}$ by replacing $\mathcal{L}_2$ with $\mathcal{L}_1$ smooth loss in (9), inside $\mathcal{L}_{symm}$ in (10) and first branch output $\hat{y}$ with the second branch output y. Supp. [1] contains more details about the loss functions.

4. Experiments

4.1. Training procedure

Datasets. Since our focus is on the human facial domain, we use FFHQ [19] dataset and 100K generated images from EG3D pre-trained on FFHQ for training and perform the evaluation on the CelebA-HQ [18, 25] test set. Supp. [1] contains more details about the dataset.

Training details. Our pre-trained EG3D generator is also trained on the FFHQ dataset [19]. We train two versions of the same model: *Ours*, trained on FFHQ and synthetic data, and *Ours (FFHQ)*, trained only on FFHQ data. We discuss the motivation to use synthetic samples in Sec. 4.2 and describe the training procedure details in Supp. [1].

Table 2. Quantitative comparison on save-view inversion. The inference time, including the EG3D pass, is given for a single RTX A100 Ti GPU. *Ours* exceeds other encoder-based methods by photometric scores and embeds the head in 3D space significantly better than all the other methods (see *Depth* ↓).

	Method	MSE↓	LPIPS↓	ID ↑	MS-SSIM↑	Depth↓	Infer. time ↓
Optim.	$\mathcal{W}+$ [20]	0.071	0.17	0.55	0.80	0.086	**77.07** s
	PTI [37]	0.013	0.07	0.76	0.89	0.087	119.34 s
	P. Opt. [21]	0.014	0.08	0.67	0.88	0.119	110.86 s
	SPI [49]	**0.005**	**0.05**	**0.94**	**0.95**	**0.078**	258.84 s
Encoder	e4e [42]	0.060	0.21	0.33	0.70	0.061	**0.04 s**
	pSp [36]	0.045	0.18	0.40	0.73	0.076	**0.04 s**
	EG3D-GOAE [50]	0.026	0.11	0.67	0.84	0.053	0.18 s
	Ours *(FFHQ)*	0.016	0.07	**0.78**	0.89	**0.042**	0.12 s
	Ours	**0.015**	**0.06**	0.77	**0.90**	0.047	0.12 s

Table 3. Quantitative comparison on novel view rendering of the inverted representation. We outperform all the other baselines on extreme novel view yaw angles. MSE and others are not suitable here due to spatial misalignment.

	Method	ID ↑ for novel yaw angle (rad): -0.8	-0.6	-0.3	0.3	0.6	0.8	ID ↑ Avg
Optim.	$\mathcal{W}+$ [20]	0.26	0.32	0.42	0.42	0.33	0.28	0.33
	PTI [37]	0.34	0.41	0.55	0.56	0.43	0.35	0.44
	P. Opt. [21]	0.30	0.36	0.50	0.50	0.38	0.31	0.39
	SPI [49]	**0.43**	**0.52**	**0.70**	**0.71**	**0.54**	**0.44**	**0.55**
Encoder	e4e [42]	0.19	0.23	0.28	0.28	0.24	0.21	0.23
	pSp [36]	0.24	0.28	0.35	0.36	0.29	0.25	0.29
	EG3D-GOAE [50]	0.38	0.46	0.57	0.57	0.48	0.40	0.47
	Ours *(FFHQ)*	**0.45**	**0.54**	**0.66**	**0.67**	**0.56**	**0.47**	**0.55**
	Ours	0.44	0.53	**0.66**	**0.67**	0.55	0.46	**0.55**

Baselines. We compare our approach with both optimization- and encoder-based inversion methods. Among optimization-based methods, we compare to universal $\mathcal{W}+$ optimization [20] and PTI [37], as well as to Pose Opt. [21] and SPI [49], recently introduced for 3D GANs. Among encoder-based methods, we compare to e4e [42], pSp [36] and EG3D-GOAE [50]. For $\mathcal{W}+$ optimization, we optimize the latent code for 1K steps. For PTI, we first optimize the latent code $\hat{w} \in \mathcal{W}+$ for 1K steps and then fine-tune the generator for 1K steps. For Pose Opt. and SPI, we re-run their official implementation. For pSp, we employ the original training configuration from [36] with a batch size of 3. We train the pSp encoder on both FFHQ and synthetic data, similarly to our method. For EG3D-GOAE, we take the released checkpoint and run the inference on our dataset.

4.2. Results

Comparison to the state-of-the-art. We present the evaluation of our approach w.r.t. the baselines in Fig. 3 and Table 2. Commonly used metrics MSE, LPIPS [52], MS-SSIM [45], and ID similarity [17] (measured by the pre-trained face recognition network not used in training) have been selected to analyze various aspects of perceptual sim-

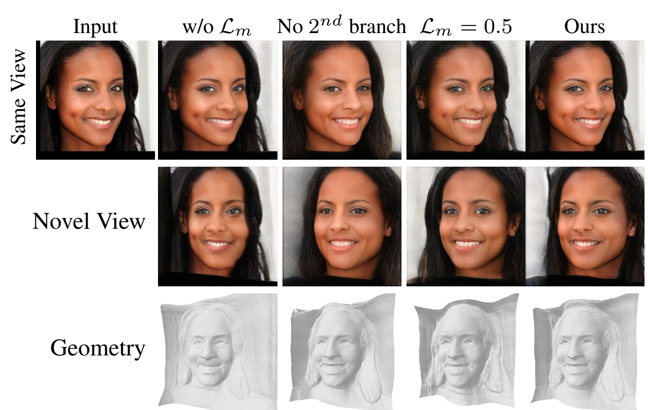

Figure 5. Qualitative ablation study for the loss, dataset, and architecture changes. *Electronic zoom-in recommended.*

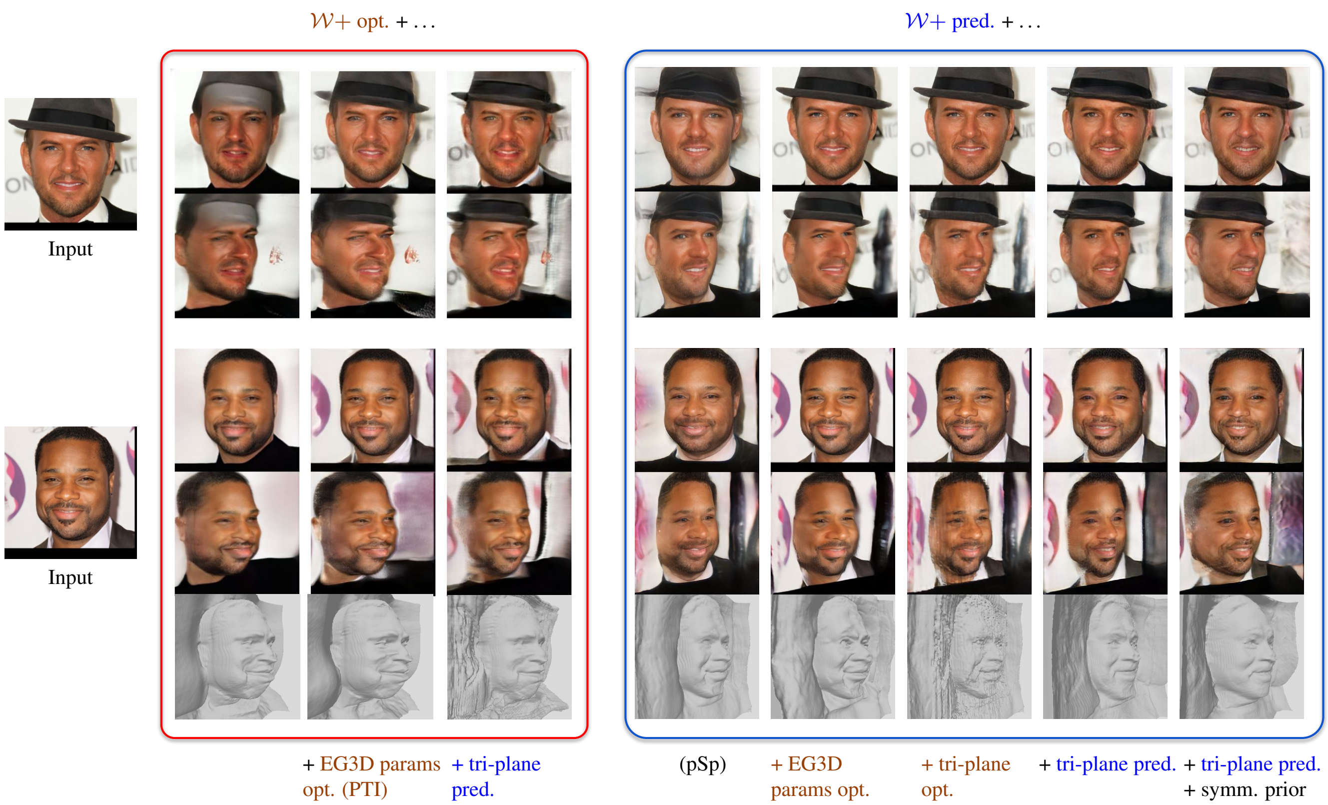

Figure 6. Comparison of hybrid approaches on CelebA-HQ test dataset. We refer to the computation done via optimization as opt. and via an encoder as pred. We observe that the methods starting from $\mathcal{W}+$ opt. yield elongated head geometry, whereas subsequent tri-plane pred. can partially alleviate it. Experiments starting from $\mathcal{W}+$ pred. demonstrate that the tri-plane space is more spatially restrictive than the EG3D parameters space. *Ours* = $\mathcal{W}+$ pred. + tri-plane pred. + symmetry prior. *Electronic zoom-in recommended.*

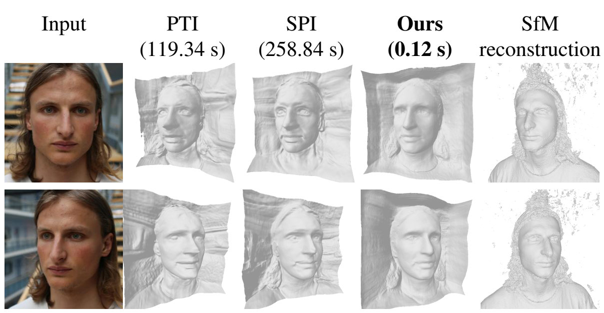

Figure 7. Comparison of the estimated 3D geometry w.r.t. the "ground-truth" reconstruction by Structure-from-Motion (SfM). Our method estimates the view-consistent embedding of a head in 3D from a single image. *Electronic zoom-in recommended.*

ilarity between inputs and corresponding reconstructions. To assess the quality of 3D geometry, we measure Depth MSE following a similar protocol from [39]. Among optimization-based techniques, SPI achieves the best results for same-view reconstruction. We attribute this to the almost three times longer optimization time than PTI and Pose Opt. However, our method outperforms all the baselines according to the Depth metric while being an order of magnitude times faster than optimization-based approaches. Effectively, this means that the head shape is closer to the one estimated by a parametric face prior model than for the other methods.

Furthermore, we demonstrate the identity preservation quality of input image re-rendering from a novel view in Table 3 and in Fig. 4. We outperform all the baselines on extreme novel view yaw angles, and our method embeds the head in 3D space in a much more plausible way while preserving fine details and not relying on any explicit face or head priors. For all methods in general, the ID score declines faster above a certain value of the yaw angle due to the uneven angle distribution in the FFHQ dataset.

Ablation study. In Fig. 5 and Table 4, we ablate over the possible differences in our model design, such as loss functions weights and the presence of the second branch. As some of those were introduced to handle occluded regions in the input view, we demonstrate visually how the incorporated symmetric prior affects the novel view and 3D geometry quality. All models in this ablation except the first branch encoder are trained for 600K steps. We observed that including symmetric prior significantly improves novel-view quality and geometric consistency. De-

Table 4. Quantitative ablation study for the loss, dataset, and architecture changes.

Method	MSE ↓	LPIPS ↓	MS-SSIM ↑	Depth ↓	ID ↑						
					Same View	Novel View (Yaw angle in radians)					
						-0.8	-0.6	-0.3	0.3	0.6	0.8
Ours	0.019	**0.08**	0.87	0.051	0.68	0.39	0.47	0.58	0.59	0.48	0.40
… *(FFHQ)*	0.022	0.09	0.86	**0.044**	**0.70**	**0.41**	**0.48**	**0.60**	**0.60**	**0.50**	**0.42**
… w/o $\mathcal{L}_m$	**0.017**	**0.08**	**0.88**	0.082	0.69	0.39	0.47	0.59	0.59	0.48	0.40
… $\mathcal{L}_m = .005$	0.018	**0.08**	0.87	0.068	0.69	0.38	0.46	0.59	0.60	0.48	0.40
… $\mathcal{L}_m = 0.5$	0.028	0.11	0.83	0.049	0.64	0.37	0.44	0.55	0.56	0.45	0.37
No 2^{nd} branch	0.047	0.18	0.73	0.056	0.41	0.25	0.30	0.36	0.37	0.31	0.26
… *(FFHQ)*	0.047	0.18	0.74	0.051	0.44	0.28	0.33	0.39	0.40	0.34	0.29
… w/o $\mathcal{L}_m$	0.045	0.18	0.73	0.076	0.40	0.24	0.28	0.35	0.36	0.29	0.25
… $\mathcal{L}_m = .005$	0.045	0.18	0.73	0.068	0.39	0.24	0.28	0.34	0.36	0.30	0.25
… $\mathcal{L}_m = 0.5$	0.057	0.20	0.70	0.053	0.41	0.25	0.29	0.37	0.37	0.30	0.25

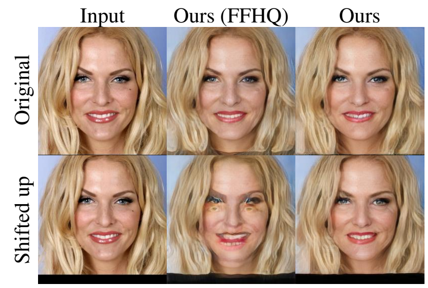

Figure 8. Adding synthetic samples to training (*Ours*) makes the model robust to the input image shifts compared to trained on FFHQ only (*Ours (FFHQ)*).

spite the model trained on FFHQ achieving a higher ID score, in Fig. 8, we show that adding synthetic EG3D samples in training allows us to correctly canonicalize face for a slightly shifted input, while our model trained with only real images is very sensitive to minor image misalignment.

Geometry evaluation for a multi-view sequence. In Fig. 7, we present a qualitative evaluation of the geometry obtained using our method against PTI [37], SPI [49], and ground truth geometry obtained using a third-party SfM software [4]. We notice that the estimated geometry is highly view-consistent and is closer to the ground truth.

We present more evaluations and visual results in the Supp. [1], such as novel view synthesis for a talking head video and face manipulation capabilities.

4.3. PTI and tri-plane offsets behavior

Both our method and optimization- and encoder-based baselines can be decomposed into two stages: estimating the latent code and the delta for the generator parameters. In Fig. 6, we show how combining these steps, each performed either by optimization (opt.) or an encoder (pred.), influences the inversion behavior.

$\mathcal{W}+$ opt. inverts a single image and cannot account for 3D geometry due to the lack of supervision from other views, which results in incorrectly stretched geometry. Accordingly, the same happens with PTI = ($\mathcal{W}+$ opt. + EG3D params opt.) method. Interestingly, tri-plane prediction, applied on top of $\mathcal{W}+$ opt., can alleviate the damage to the geometry caused by $\mathcal{W}+$ opt.

$\mathcal{W}+$ pred. by a pSp encoder, on the contrary, embeds the head in 3D more plausibly due to the supervision from images under different poses during training. At the same time, the same-view quality is marginally worse than PTI. Applying the PTI's second step (EG3D params opt.) helps improve it significantly; however, it incorrectly modifies head proportions, similar to the $\mathcal{W}+$ opt. behavior. To investigate this effect further, instead of optimizing EG3D parameters after $\mathcal{W}+$ pred., we try optimizing the tri-plane offsets directly, and this fully cancels the incorrect stretching of geometry while preserving high fidelity in the same view. Since both EG3D params opt. and tri-plane opt. are performed for a single image (i.e. without multi-pose supervision during training), this may indicate that offsetting the tri-planes is more spatially restrictive and thus stable. Therefore, we base our method on directly leveraging the tri-plane representation.

We further improve the checkerboard artifacts in novel view, noticeable for tri-plane opt., by tri-plane prediction, and improve the embedding in 3D space by a symmetric prior.

5. Conclusion

We present a novel approach for EG3D inversion that achieves high-quality reconstructions with view consistency and can be run in close to real time on modern GPUs. We also show that directly utilizing tri-plane representation better estimates 3D structure compared to other approaches while preserving identity in the novel view. Although our method achieves compelling results and is on par with optimization-based approaches, both visually and quantitatively, it has certain limitations. For instance, it is limited by the range of yaw angles shown to EG3D during training and cannot model the background depth. In addition, there is room for improvement of the temporal consistency and for supporting input images with extreme head poses.

Acknowledgments

This work was supported by the ERC Starting Grant Scan2CAD (804724) and the German Research Foundation (DFG) Research Unit "Learning and Simulation in Visual Computing." We also thank Yawar Siddiqui, Guy Gafni, Shivangi Aneja, and Simon Giebenhain for participating in the sample videos and helpful discussions.

References

[1] Triplanenet: An encoder for eg3d inversion. Supplemental material, 2024.

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

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

[4] Agisoft. *Metashape software. https://www.agisoft.com/.*

[5] 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.

[6] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[7] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022.

[8] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

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

[11] Jiankang Deng, J. Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2018.

[12] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11389–11398, 2022.

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

[14] 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.

[15] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019.

[16] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE international conference on computer vision*, pages 2439–2448, 2017.

[17] Y. 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. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5900–5909, 2020.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2017.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018.

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

[21] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2967–2976, 2023.

[22] Yushi Lan, Xuyi Meng, Shuai Yang, Chen Change Loy, and Bo Dai. Self-supervised geometry-aware encoder for style-based 3d gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20940–20949, 2023.

[23] Yu-Jhe Li, Tao Xu, Bichen Wu, Ningyuan Zheng, Xiaoliang Dai, Albert Pumarola, Peizhao Zhang, Peter Vajda, and Kris Kitani. 3d-aware encoding for style-based neural radiance fields. *arXiv preprint arXiv:2211.06583*, 2022.

[24] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017.

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2014.

[26] Junyu Luo, Yong Xu, Chenwei Tang, and Jiancheng Lv. Learning inverse mapping by autoencoder based generative adversarial nets. In *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24*, pages 207–216. Springer, 2017.

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

[28] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. *arXiv preprint arXiv:2212.01206*, 2022.

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

[30] Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in neural information processing systems*, 33:6767–6778, 2020.

[31] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

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

[33] Mohit Mendiratta Pan, Mohamed Elgharib, Kartik Teotia, Ayush Tewari, Vladislav Golyanik, Adam Kortylewski, Christian Theobalt, et al. Avatarstudio: Text-driven editing of 3d dynamic human head avatars. *arXiv preprint arXiv:2306.00547*, 2023.

[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[35] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

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

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

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

[39] Yichun Shi, Divyansh Aggarwal, and Anil K. Jain. Lifting 2d stylegan for 3d-aware face generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6254–6262, 2020.

[40] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019.

[41] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[42] 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.

[43] Alex Trevithick, Matthew A. Chan, Michael Stengel, Eric Chan, Chao Liu, Zhiding Yu, S. Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano. Real-time radiance fields for single-image portrait view synthesis. *ArXiv*, abs/2305.02310, 2023.

[44] Tengfei Wang, Bo Zhang, Ting Zhang, Shuyang Gu, Jianmin Bao, Tadas Baltrusaitis, Jingjing Shen, Dong Chen, Fang Wen, Qifeng Chen, et al. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *arXiv preprint arXiv:2212.06135*, 2022.

[45] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.

[46] Shangzhe Wu, C. Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2019.

[47] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[48] Jiaxin Xie, Hao Ouyang, Jingtan Piao, Chenyang Lei, and Qifeng Chen. High-fidelity 3d gan inversion by pseudo-multi-view optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2023.

[49] Fei Yin, Yong Zhang, Xuan Wang, Tengfei Wang, Xiaoyu Li, Yuan Gong, Yanbo Fan, Xiaodong Cun, Ying Shan, Cengiz Oztireli, et al. 3d gan inversion with facial symmetry prior. *arXiv preprint arXiv:2211.16927*, 2022.

[50] Ziyang Yuan, Yiming Zhu, Yu Li, Hongyu Liu, and Chun Yuan. Make encoder great again in 3d gan inversion through geometry and occlusion-aware encoding. *arXiv preprint arXiv:2303.12326*, 2023.

[51] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.

[52] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

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

Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16, pages 592–608. Springer, 2020.

[54] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 597–613. Springer, 2016.