Real-Time Radiance Fields for Single-Image Portrait View Synthesis

ALEX TREVITHICK*, University of California San Diego, USA
MATTHEW CHAN and MICHAEL STENGEL, NVIDIA, USA
ERIC R. CHAN*, Stanford University, USA
CHAO LIU, ZHIDING YU, and SAMEH KHAMIS, NVIDIA, USA
MANMOHAN CHANDRAKER and RAVI RAMAMOORTHI, University of California San Diego, USA
KOKI NAGANO, NVIDIA, USA

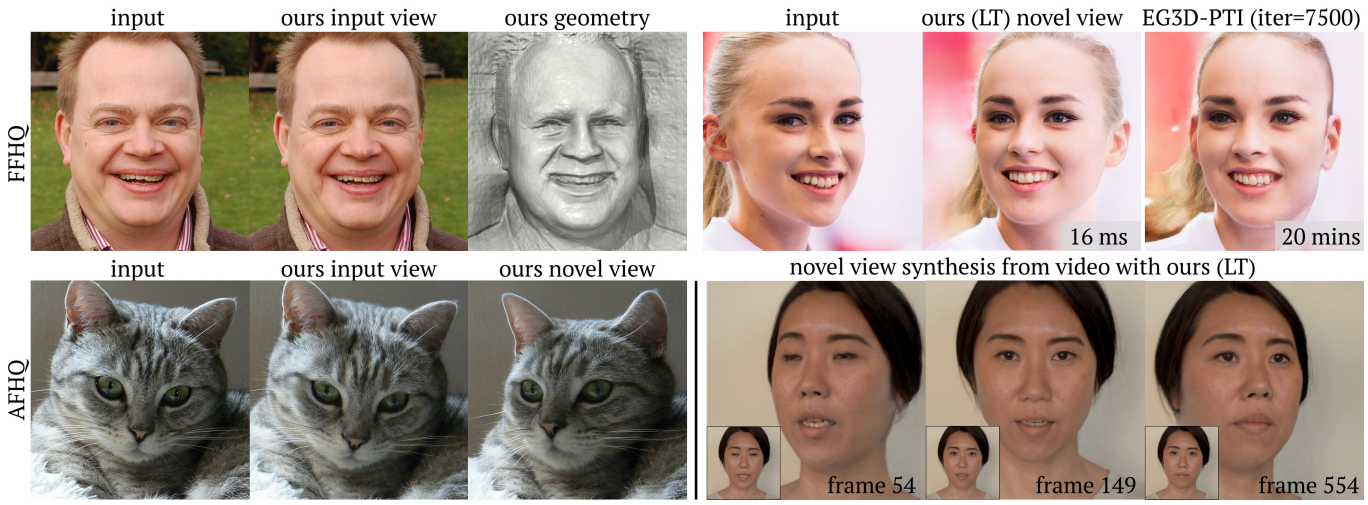

Fig. 1. Given a single RGB input image, our method generates 3D-aware images and geometry of an object (e.g., faces [top row] and cats [bottom row, left]) in real-time, while the state-of-the-art 3D GAN inversion [Chan et al. 2022] does not generate a satisfactory result after 20 mins of fine-tuning [Roich et al. 2021] (top right). Our method can also be applied to a video frame-by-frame for video-based novel view synthesis (bottom row, right). Ours (LT) refers to a lightweight faster version of our model that has almost the same quality as the full model. Credits to Erik (HASH) Hersman and 2017 Canada Summer Games.

We present a one-shot method to infer and render a photorealistic 3D representation from a single unposed image (e.g., face portrait) in real-time. Given a single RGB input, our image encoder directly predicts a canonical triplane representation of a neural radiance field for 3D-aware novel view synthesis via volume rendering. Our method is fast (24 fps) on consumer hardware, and produces higher quality results than strong GAN-inversion baselines that require test-time optimization. To train our triplane encoder pipeline, we use only synthetic data, showing how to distill the knowledge from a pretrained 3D GAN into a feedforward encoder. Technical contributions include a Vision Transformer-based triplane encoder, a camera data augmentation strategy, and a well-designed loss function for synthetic data training. We benchmark against the state-of-the-art methods, demonstrating significant improvements in robustness and image quality in challenging real-world settings. We showcase our results on portraits of faces (FFHQ) and cats (AFHQ), but our algorithm can also be applied in the future to other categories with a 3D-aware image generator.

CCS Concepts: • **Computing methodologies** → **Image-based rendering**.

Additional Key Words and Phrases: View Synthesis, Inverse Rendering, Neural Radiance Field

ACM Reference Format:
Alex Trevithick, Matthew Chan, Michael Stengel, Eric R. Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano. 2023. Real-Time Radiance Fields for Single-Image Portrait View Synthesis. *ACM Trans. Graph.* 1, 1, Article 1 (January 2023), 15 pages. https://doi.org/10.1145/3592460

1 INTRODUCTION

Digitally reproducing the 3D appearance of an object from a single image is a long-standing goal for computer graphics and vision. Interactive synthesis of photorealistic novel views opens new possibilities for AR/VR, and for 3D telepresence and videoconferencing when applied to humans. In this work, we propose a technique to

*This project was initiated and substantially carried out during an internship at NVIDIA.

Authors' addresses: Alex Trevithick, University of California San Diego, La Jolla, USA; Matthew Chan; Michael Stengel, NVIDIA, Santa Clara, USA; Eric R. Chan, Stanford University, Stanford, USA; Chao Liu; Zhiding Yu; Sameh Khamis, NVIDIA, Santa Clara, USA; Manmohan Chandraker; Ravi Ramamoorthi, University of California San Diego, La Jolla, USA; Koki Nagano, NVIDIA, Santa Clara, USA.

0730-0301/2023/1-ART1
https://doi.org/10.1145/3592460

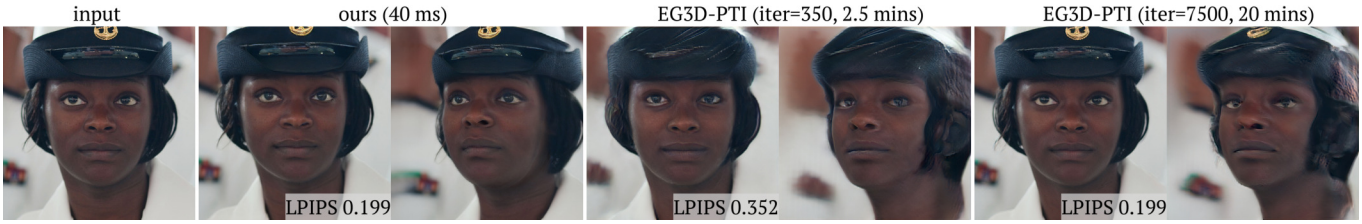

Fig. 2. Comparison to the state-of-the-art 3D GAN[Chan et al. 2022] with test-time fine tuning[Roich et al. 2021] (EG3D-PTI). Single-view 3D GAN inversion approaches trade off the 2D reconstruction quality and the 3D effects. When fine tuned longer (7500 iterations), EG3D-PTI can capture the same fine-scale details as ours ($LPIPS = 0.199$), but the quality of another view starts to degrade. On the other hand, our method captures out-of-domain details (e.g., emblem) in one-shot while producing realistic rendering of another view, and operating in real-time. Credit to Obama White House.

infer a 3D representation for real-time view synthesis given a single portrait-style input image (e.g., of a human face, see Fig. 1).

Recently, 3D aware-image generation approaches (e.g., [Chan et al. 2022; Deng et al. 2022; Skorokhodov et al. 2022]) demonstrated unconditional generation of photorealistic 3D representations from a collection of single-view 2D images by combining NeRF-based representations [Mildenhall et al. 2020] and GANs [Goodfellow et al. 2014]. Notably, EG3D [Chan et al. 2022] proposed an efficient triplane 3D representation and demonstrated real-time 3D-aware image rendering with quality comparable to 2D GANs. Once trained, the 3D GAN generators can be frozen and used for single-image 3D reconstruction tasks via GAN inversion [Karras et al. 2020] and test-time fine tuning [Roich et al. 2021]. However, there are a few challenges in this 3D-GAN inversion-based methods. (1) Due to the multi-view nature of training a NeRF, it needs careful optimization objectives and additional 3D priors [Xie et al. 2022a; Yin et al. 2022] in the single view setting to avoid unsatisfactory results on novel views and corrupted geometry (see Fig. 6). Fig. 2 shows the tradeoff in the SOTA single-view 3D GAN inversion pipeline. (2) The test-time optimization requires an accurate camera pose as input or to be jointly optimized [Ko et al. 2023]. (3) The above optimization for every single image is time-consuming, limiting the technique for real-time video applications.

In this paper, we present a one-shot approach to lift an input 2D portrait image to 3D in real-time (24fps on consumer hardware, see Tab. 1). Unlike previous work that reuses a pre-trained generator, we train an encoder end-to-end that directly predicts the triplane 3D features from a single input image. In contrast to prior works that use multiview real image acquisition setups, we do not need any real images at all, nor do we require time-consuming physically-based rendering of high-quality and expensive face assets.

Instead, we fully supervise the training of our triplane encoder for novel view synthesis using multiview-consistent synthetic data generated from a pre-trained 3D GAN. Together with our data augmentation strategies and Transformer-based encoder, we present a model which can handle challenging real-world input images including occlusion and three-quarter views. We showcase our results on human and cat face categories in this paper, but the methodology can apply to any category for which 3D-aware image generators are available. Our work may motivate applications such as temporally consistent view synthesis; Fig. 1 (bottom right) shows our method applied to a video in a frame-by-frame fashion without any special handling.

Table 1. Time taken to lift the input image to 3D (Encoding) and render (Render) a 3D representation given an input image on a single RTX 3090 GPU. The end-to-end runtime with our model and our lightweight model (LT) is significantly faster than NeRF-based baselines. †ROME employs 2D-based neural rendering with mesh-based neural textures, producing the output at 256x256 resolution; it also requires a segmentation mask and detected keypoints from off-the-shelf models which requires around 200ms.

Time	H.NeRF	ROME	EG3D-PTI	Ours	Ours (LT)
Encoding	60s	60ms†	2 mins	**40ms**	**16ms**
Render	58ms	31ms	24ms	24ms	24ms

In summary, contributions of our work include:

- We propose a feed forward encoder model to directly infer a triplane 3D representation from an input image. No test-time optimization is needed.
- We present a new strategy for training a feed forward triplane encoder for 3D inversion using *only* synthetic data generated from a pre-trained 3D-aware image generator.
- We demonstrate that our method can infer a photorealistic 3D representation in real-time given a single *unposed* image. Together with our Transformer-based encoder and on-the-fly augmentation strategy, our method can robustly handle challenging input images of side views and occlusions.

2 RELATED WORK

Our work touches on light fields, few-shot view synthesis, learning with synthetic data, 3D-aware portrait generation, and GAN inversion. Our focus is on real-time view synthesis from a single image, and we do not address portrait relighting or editing. Tab. 1 summarizes runtime for inferring 3D representations from an input and rendering. Our one-shot method is three orders of magnitude faster than the NeRF-based state-of-the-art methods for inference, enabling a real-time pipeline.

Light Fields and Image-Based Rendering. View synthesis or image-based rendering has a long history in computer graphics and vision [Chen and Williams 1993; McMillan and Bishop 1995], and has often been framed in terms of reconstructing the light field [Gortler

et al. 1996; Levoy and Hanrahan 1996]. However, those methods typically required hundreds of views. Subsequent light field approaches demonstrated few-shot general [Kalantari et al. 2016] and even single image view synthesis for categories [Srinivasan et al. 2017], but required light field camera training data. More recently, neural-field based approaches [Mildenhall et al. 2020; Xie et al. 2022b] combine recent neural implicit 3D representations [Chen and Zhang 2019; Mescheder et al. 2019; Park et al. 2019; Sitzmann et al. 2019] with volume rendering for novel view synthesis, but require a large number of input images per scene.

Few-shot novel view synthesis. Some recent work extends NeRF for training from even a single view [Xu et al. 2022a] or for few-shot novel view synthesis using fully implicit 3D representations [Jang and Agapito 2021; Li et al. 2022; Trevithick and Yang 2021; Yu et al. 2021], 3D convolutions [Chen et al. 2021; Yu et al. 2022], or Transformers [Lin et al. 2023; Wang et al. 2021b]. However these approaches do not generate novel views in real-time. Moreover, all of the above approaches need multi-view images to train their models. Our method, on the other hand, only needs synthetic images generated from a pre-trained 3D GAN, which is trained by a collection of single-view images. FWD [Cao et al. 2022] builds on top of SynSin[Wiles et al. 2020] for real-time novel view synthesis using depth-based image warping, but requires depth data from multi-view stereo or a depth sensor. Yet another family of approaches is the geometry-free method[Ren and Wang 2022; Rombach et al. 2021; Sajjadi et al. 2022], but they need a large number of images to learn precise ray transformations; otherwise it may lead to blurry or multiview inconsistent results.

Learning with synthetic data. Synthetic data provides useful supervision for training a deep learning model when ground truth data is not available. Previous methods used synthetic data for various deep learning-based tasks such as dense visual alignment [Peebles et al. 2022], 3D face reconstruction [Pan et al. 2021; Wood et al. 2022] and analysis [Wood et al. 2021], portrait normalization [Nagano et al. 2019; Zhang et al. 2020], and semantic segmentation [Tritrong et al. 2021; Zhang et al. 2021]. Some previous work used synthetic face portrait images generated by rendering 3D face assets using a phycally-based pathtracer to train a model for portrait relighting [Yeh et al. 2022] or relighting and view synthesis [Sun et al. 2021]. Since the CG rendering exhibits a synthetic look, they need an additional step to adapt to real images. Other concurrent work [Ko et al. 2023] uses a discrete number of pre-generated synthetic images from a 3D-aware generator [Chan et al. 2022] for 3D GAN inversion tasks. Instead, we generate an unlimited amount of synthetic data in the training loop and show that on-the-fly camera augmentation is critical for generalization to real images for synthetic data training.

3D-aware portrait generation and manipulation. For a well-known category of object, such as human faces, previous work [Athar et al. 2022; Gao et al. 2020; Groueix et al. 2018; Hong et al. 2022a; Khakhulin et al. 2022; Kim et al. 2018; Mihajlovic et al. 2022; Nagano et al. 2018; Wang et al. 2022a] used 3D face priors for few-shot portrait synthesis. While the face priors provide additional capabilities for facial manipulations and expression retargeting [Seol et al. 2011], they don't generalize beyond humans. Recently, 3D aware-image generation approaches [Chan et al. 2021; Nguyen-Phuoc et al. 2019; Niemeyer and Geiger 2021; Schwarz et al. 2020] started to tackle the problem of unconditional generation of photorealistic 3D representations from a collection of single-view 2D images. By combining neural volumetric rendering[Mildenhall et al. 2020] and generative adversarial networks (GANs)[Goodfellow et al. 2014], recent 3D GAN approaches[Chan et al. 2022; Deng et al. 2022; Gu et al. 2021; Or-El et al. 2022; Rebain et al. 2022; Skorokhodov et al. 2022; Xiang et al. 2022; Xu et al. 2022b; Zhang et al. 2022; Zhou et al. 2021] started to demonstrate an ability to generate high-resolution multi-view consistent images and geometry of a category of objects. We adapt the efficient triplane 3D representation from EG3D [Chan et al. 2022] and demonstrate single-view novel view synthesis on similar categories.

3D GAN inversion. Following the success of GAN inversion in 2D domains for image editing and manipulations[Alaluf et al. 2021; Dinh et al. 2022; Richardson et al. 2021; Tov et al. 2021; Wang et al. 2022c], existing 3D GAN inversion methods [Ko et al. 2023; Lin et al. 2022; Sun et al. 2022] project a given image to variants of the pre-trained StyleGAN2 latent space [Abdal et al. 2019; Karras et al. 2020]. Assuming multiview images, FreeStyleGan [Leimkühler and Drettakis 2021] proposes to map projected camera coordinates to a subject-specific StyleGAN2 latent space which allows the subject to be rendered from specified cameras under the constraints of the StyleGAN prior. While this global latent space provides an additional ability for 3D-aware portrait *editing*, the StyleGAN2 latent space trades off reconstruction fidelity for editability, making the exact reconstruction of the input image challenging. Thus, existing 3D GAN inversion approaches require an approximate camera pose and slight generator weight tuning [Feng et al. 2022; Roich et al. 2021] at test time to reconstruct out-of-domain input images. Our feed forward encoder takes an unposed image as input and does not need test-time optimization for camera poses unlike concurrent work [Ko et al. 2023].

Talking-head generators. Given a single target portrait and a driving video, recent talking-head generators can reenact the portrait by transferring facial expressions and head poses from the driver video [Doukas et al. 2021; Drobyshev et al. 2022; Hong et al. 2022b; Wang et al. 2021a, 2022b; Zakharov et al. 2020; Zhao and Zhang 2022]. Trained by video datasets, their methods mainly focuse on talking-head video generation by manipulating avatar poses and expressions within a 2D portrait. As such, they do not predict volumetric representations that allow free viewpoint rendering including background and do not provide dense 3D geometry like our method. Therefore, we do not compare to these approaches.

3 PRELIMINARIES: TRIPLANE-BASED 3D GAN

We first give an overview of the state-of-the-art 3D GAN method, EG3D, [Chan et al. 2022] from which our method will distill knowledge. EG3D learns unconditional 3D-aware image generation from a collection of single-view images and corresponding noisy camera poses, where each image has resolution 512×512. As mentioned in Sec. 2, EG3D makes use of a hybrid triplane representation to

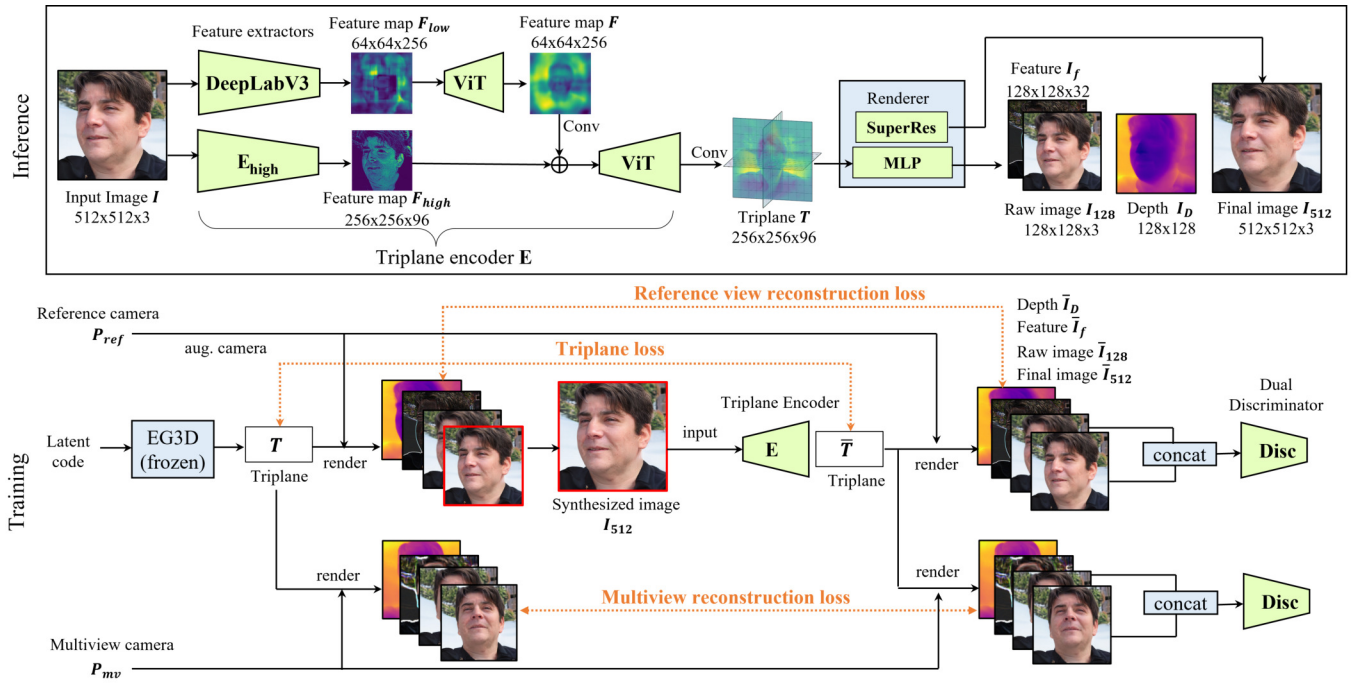

Fig. 3. Inference and training outline for our pipeline. At inference, we take an unposed image, and extract low resolution features F_{low} with a DeepLabv3 backbone. These features are fed to a ViT yielding F and then concatenated after convolution with high-resolution features F_{high} before being decoded with a ViT and convolutions to a triplane representation. These features condition the volumetric rendering process which yields depth, feature, color, and superresolved images. During training, we sample an identity from EG3D and then render two supervision views. The first serves as the input to our encoder, which predicts a triplane, which conditions volume rendering from the same two views. The rendering results are compared with those of EG3D as outlined in Sec. 4. Feature maps are visualized for illustration.

condition the neural volumetric rendering process, whereby three 2D feature grids are stored along each of the three canonical planes– xy, xz, yz. Using a StyleGAN2 generator [Karras et al. 2020], EG3D maps a noise vector and conditioning camera poses to a triplane representation $\boldsymbol{T} \in \mathbb{R}^{256\times256\times96}$ which corresponds to the 3 axis-aligned planes, each with 32 channels. These features condition the neural volumetric rendering.

To assign a point $\boldsymbol{x} \in \mathbb{R}^3$ with its feature, color and volume density, $(\mathbf{f}, \mathbf{c}, \sigma)$, a lightweight MLP decodes the three feature vectors gathered by projecting $\boldsymbol{x}$ to each of the canonical planes:

$$(\mathbf{f}, \mathbf{c}, \sigma) = \text{MLP}(\Phi(\boldsymbol{f}_{xy}, \boldsymbol{f}_{xz}, \boldsymbol{f}_{yz})), \tag{1}$$

where $\boldsymbol{f}_{ij}$ are the features gathered by projecting $\boldsymbol{x}$ to the ij plane and bilinearly interpolating the nearby features, and Φ is the mean operator. Note that output values including the color are independent of viewing direction and only depend on $\boldsymbol{x}$. By accumulating many points along rays, and performing volume rendering [Max 1995] as in NeRF [Mildenhall et al. 2020], one may render a feature image $\boldsymbol{I}_f \in \mathbb{R}^{32\times128\times128}$ and a raw neural rendering RGB image $\boldsymbol{I}_{128} \in \mathbb{R}^{3\times128\times128}$ from a given camera pose. In practice, $\boldsymbol{I}_{128}$ corresponds to the first three channels of the feature image $\boldsymbol{I}_f$.

We additionally extract a dense depth map $\boldsymbol{I}_D \in \mathbb{R}^{128\times128}$ from this volume rendering, which we use later to supervise our model. The neural rendered images $\boldsymbol{I}_{128}$ and $\boldsymbol{I}_f$ are then fed to a 2D super-resolution network, which yields the final superresolved rendering output:

$$\text{SuperRes}(\boldsymbol{I}_f, \boldsymbol{I}_{128}) = \boldsymbol{I}_{512} \in \mathbb{R}^{3\times512\times512}. \tag{2}$$

This 3D GAN pipeline is trained end-to-end following 2D GAN training with a 2D (dual) discriminator. The reader is referred to the original paper [Chan et al. 2022] for full details.

The efficient design of EG3D allows rendering from a triplane at 42 fps on the RTX 3090. At the same time, EG3D provides comparable quality to even the state-of-the-art 2D GANs by FID. These attributes provide a strong basis for supervising our encoder-based method using EG3D-generated synthetic data.

4 METHOD

Our goal is to distill the knowledge of a fully trained EG3D generative model (learned over a category or set of categories) into a feedforward encoder pipeline that can *directly map an unposed image to a canonical triplane 3D representation** *which can be decoded with a NeRF.* This pipeline requires only a single feedforward network pass, thus avoiding the expensive GAN inversion process, while allowing free viewpoint re-rendering of the input in real-time.

*Note that each category has a different notion of canonical representation: for human faces, the center of the head is the origin, and planes orthogonally intersect the head up-to-down, left-to-right, and front-to-back.

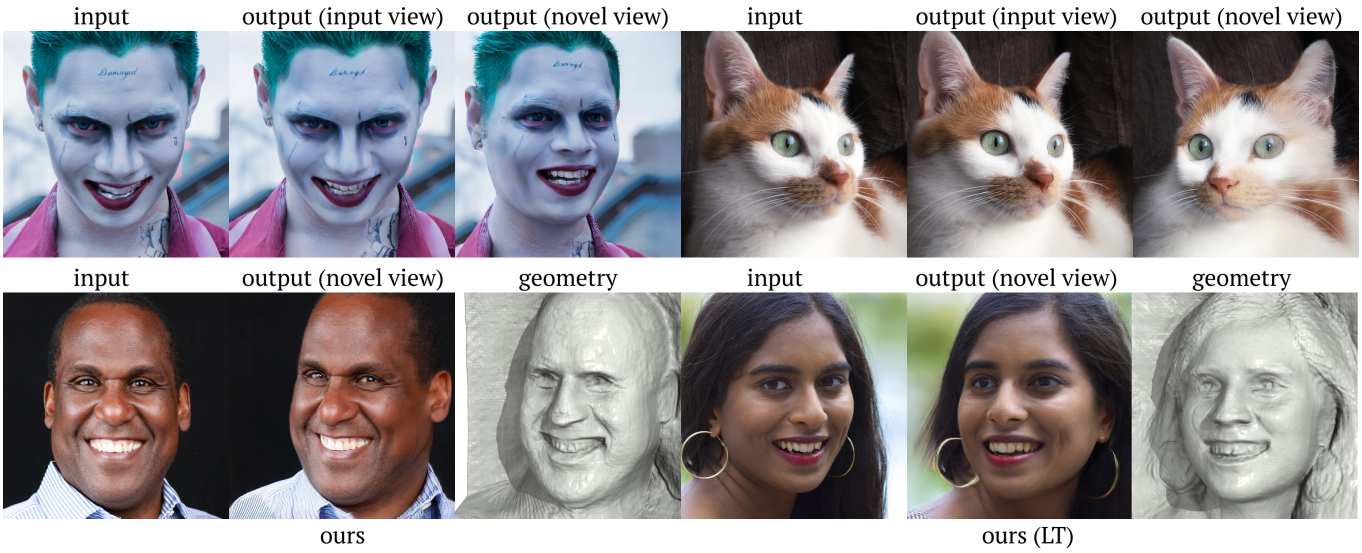

Fig. 4. FFHQ and AFHQ qualitative results from our model (left) and our lightweight model (LT) (right). We showcase reconstructed input and novel views, or the learned geometry. In the bottom-right, note our model's ability to infer structure consistent with the input behind occlusion. Credits to YuChen Cheng, Montclair Film, Lydia Liu.

Note that our contribution focuses on the image-to-triplane encoder and associated synthetic training method, as shown in the pipeline of Fig. 3. We make use of the MLP volume renderer and superresolution architectures from EG3D as per Eqns 1 and 2 and train all the components end-to-end. In Tab. 1, the top row shows that our image to triplane inference runs at up to 60 fps (16 ms), while rendering has identical performance to EG3D (bottom row of Tab. 1).

4.1 Triplane encoder

We note that inferring a canonicalized 3D reprentation (i.e., the inferred 3D representation is frontalized and aligned) from an arbitrary RGB image while simultaneously synthesizing precise subject-specific details from the input is a highly non-trivial task. We break this challenge into the two-fold goals: 1) to create a canonicalized 3D representation of the subject from an image, and 2) to render high-frequency person-specific details. We note these goals are often at odds with one another, and exemplify the bias-variance tradeoff whereby the output will resemble the input well, but may not be correctly canonicalized in 3D (see Fig. 12), or the output will have the correct 3D structure, but not resemble the 2D input image (see Fig. 11). Our encoder manages to accomplish both of these goals simultaneously. Specifically, we develop and train a hybrid convolutional-Transformer encoder, $\mathbf{E}$, which maps from an unposed RGB image, $\boldsymbol{I}$, to the *canonical* triplane representation.

As seen in the upper half of Fig. 3, the architecture of our encoder begins with a fast convolutional backbone, DeepLabV3 [Chen et al. 2017], which extracts robust low-resolution features, $\boldsymbol{F}_{\text{low}} = \text{DeepLabV3}(\boldsymbol{I})$. These features are then fed to a Vision Transformer (and CNN) which gives a global inductive bias to the intermediate output features,

$$\boldsymbol{F} = \text{Conv}(\text{ViT}(\boldsymbol{F}_{\text{low}})), \tag{3}$$

where Conv is a CNN and ViT is the Vision Transformer Block from Segformer [Xie et al. 2021] with efficient self-attention. We choose the Segformer ViT for two reasons: 1) it was designed to quickly map to a high-resolution output space similar to a triplane, and 2) the efficient self-attention mechanism allows the use of high-resolution intermediate feature maps so that all information flows from input to triplane.

We consider the ViT features as having successfully created a canonicalized 3D representation of the subject (completing the step 1 above), and found during our experimentation that this shallow encoder is sufficient to reasonably canonicalize a subject, yet cannot represent important high-frequency or subject-specific details like strands of hair or birthmarks.

In order to simultaneously complete the second step (adding high-frequency detail), we next reincorporate high-resolution image features. We convolutionally encode the image again with only a single downsampling stage with encoder $\mathbf{E}_{\text{high}}$ to obtain features $\boldsymbol{F}_{\text{high}} = \mathbf{E}_{\text{high}}(\boldsymbol{I})$. These are concatenated with the extracted global features and passed through another Vision Transformer, which is finally decoded to a triplane with convolutions as seen in Fig. 3. Thus, the output of our encoder has the following form:

$$\boldsymbol{T} = \mathbf{E}(\boldsymbol{I}) = \text{Conv}(\text{ViT}(\boldsymbol{F} \oplus \boldsymbol{F}_{\text{high}})), \tag{4}$$

where $\oplus$ denotes concatenation along the channel axis, and $\boldsymbol{T}$ is triplane feature representation used in Sec. 3.

4.2 Training with synthetic data

As seen in Fig. 3 in the training step, we train our triplane encoder with synthetic data. Sampling a latent vector and passing it through the EG3D generator yields a corresponding triplane, $\boldsymbol{T}$. Given camera

parameters $\boldsymbol{P}$ (a focal length, principal point, camera orientation and position), we can render any image from the frozen EG3D generator and $\boldsymbol{T}$. At each gradient step, we synthesize two images of the same identity (same latent code) from a reference (input) camera $\boldsymbol{P}_{\text{ref}}$ and another camera $\boldsymbol{P}_{\text{mv}}$ for multiview supervision. Using the same notation as in Sec. 3, each rendering pass will give us four images: $\boldsymbol{I}_f$, $\boldsymbol{I}_{128}$, $\boldsymbol{I}_{512}$, and $\boldsymbol{I}_D$ as seen in Fig. 3.

Again as shown in Fig. 3, the input to our encoder is the high-resolution image $\boldsymbol{I}_{512}$ (highlighted in red) rendered from the input camera $\boldsymbol{P}_{\text{ref}}$, so that $\overline{\boldsymbol{T}} = \text{E}(\boldsymbol{I}_{512})$. We then use $\overline{\boldsymbol{T}}$ to condition the volume rendering process from both camera $\boldsymbol{P}_{\text{ref}}$ and $\boldsymbol{P}_{\text{mv}}$, to get two more sets of four images, which we denote as $\overline{\boldsymbol{I}}_f$, $\overline{\boldsymbol{I}}_{128}$, $\overline{\boldsymbol{I}}_{512}$, and $\overline{\boldsymbol{I}}_D$. Our loss intuitively compares those quantities synthesized by EG3D and those created by our encoder, along with a generative adversarial objective as follows:

$$L = L_{\text{tri}} + L_{\text{col}} + L_{\text{LPIPS}} + L_{\text{feat}} + \lambda_1 L_{\text{adv}} + \lambda_2 L_{\text{cate}} \tag{5}$$

L_{tri} is the L1 loss between $\boldsymbol{T}$ and $\overline{\boldsymbol{T}}$; L_{col} is the mean L1 loss computed between both sets of pairs $(\boldsymbol{I}_{128}, \overline{\boldsymbol{I}}_{128})$ and $(\boldsymbol{I}_{512}, \overline{\boldsymbol{I}}_{512})$; L_{LPIPS} is the LPIPS perceptual loss [Zhang et al. 2018] computed over both sets of pairs $(\boldsymbol{I}_{128}, \overline{\boldsymbol{I}}_{128})$ and $(\boldsymbol{I}_{512}, \overline{\boldsymbol{I}}_{512})$; L_{feat} is the mean L1 loss computed between the pairs $(\boldsymbol{I}_f, \overline{\boldsymbol{I}}_f)$; L_{adv} is the adversarial loss using a pretrained dual discriminator from EG3D which is fine-tuned during training; λ_1 is 0.1 for the reference image or 0.025 for the multiview image; and L_{cate} is an optional category-specific loss. For human faces, we use λ_2 to be 1 with face identity features from ArcFace [Deng et al. 2019a] following practice in 2D GAN inversion [Richardson et al. 2021; Tov et al. 2021]. For cat faces, we set λ_2 to 0. This objective is optimized end-to-end, i.e., with respect to all of the parameters of the encoder, rendering and upsampling modules. Note that the rendering, upsampling, and dual discriminator modules are all fine-tuned from the pretrained EG3D. However, the dual discriminator in our pipeline doesn't rely on *any* real data; instead, we train this discriminator to differentiate between images rendered from our encoder model and images rendered from the frozen EG3D. An ablation showing its effectiveness is provided in Tab. 5 and Fig. 13.

On-the-fly augmentation. Naively optimizing this objective will yield a model which performs almost perfectly on synthetic data, but lacks the ability to generalize to real images (see Fig. 12). In order to remedy this, we augment the standard EG3D rendering method which assumes a fixed camera roll, focal length, principal point and distance from subject. In contrast, we sample all four of these values from random distributions to choose the camera parameters $\boldsymbol{P}_{\text{ref}}$. The details of these distributions for each dataset are given in the supplement. For $\boldsymbol{P}_{\text{mv}}$, we choose fixed values as in the EG3D model. For $\boldsymbol{P}_{\text{ref}}$, we sample the cameras from a pitch range of ±26° and yaw range of ±49° relative to the front of a human face. For $\boldsymbol{P}_{\text{mv}}$, we sample the cameras from a pitch range of ±26° and yaw range of ±36° relative to the front of a human face. This allows the supervision of our model to happen with highly variable camera poses, forcing the model to learn to effectively canonicalize and infer from challenging images as seen in Fig. 4.

Implementation details. Before training with the full adversarial objectives in Eqn. 5, we warm up the model by training over 30k iterations without the adversarial loss and continue to train the model with the full loss functions in Eqn. 5 over 220k iterations. Since we sample two camera poses per iteration (with batch size 32), we effectively use over 16 million images during the training, which is not obtainable from real images (nor even physically-based rendered images) in practice. For full implementation details, please refer to the supplement. We train two encoders with two different compute budgets: "Ours", which has 87M parameters and "Ours (LT)", a lightweight model (LT) which has 63M parameters. The main difference between the two is in resolution of the intermediate feature maps, which result in fewer parameters in the LT model, but both contain the same structure outlined above. "Ours" runs in 22ms on a single A100 GPU (where rendering takes 15ms) and 40ms on RTX 3090 as seen in Table 1. "Ours (LT)" runs in just 16ms on RTX 3090, while retaining strong performance (see the numerical evaluations in Tabs. 2 and 3). Figures 1 and 4 show the qualitative outputs from both models.

5 RESULTS

We evaluate methods for single-view novel view synthesis on 3 main aspects (1) 2D image reconstruction (LPIPS [Zhang et al. 2018], DISTS [Ding et al. 2022], SSIM [Wang et al. 2004]) and likeness (identity consistency) (2) general image quality (FID [Heusel et al. 2017]) and (3) 3D reconstruction quality (depth, and pose estimation). For the reconstruction tasks, we need to re-render our outputs to the input views for the purpose of the evaluation using a camera pose estimated using an off-the-shelf pose predictor [Deng et al. 2019b]. However, we noticed that errors present in the estimated poses create a small image misalignment between the ground truth and our feedforward results (as opposed to inversion models which directly optimize for the given view), making the raw pixel metrics like PSNR and SSIM unreliable. For this reason, we mainly rely on the deep perceptual image metrics such as LPIPS and DISTS, which judge that the given images are of the same perceptual quality for our evaluation. Nonetheless, we report SSIM results in the main paper and include PSNR results in the supplement along with an analysis of alignment issues. In the end, our experiments qualitatively and quantitatively support that our method achieves the state-of-the-art results on in-the-wild portraits as well as multiview 3D scan datasets. For more results, please refer to the supplement video.

Datasets. Our method is evaluated on FFHQ [Karras et al. 2019], a representative dataset for high-quality in-the-wild human portraits, H3DS [Ramon et al. 2021], which has high resolution ground truth 3D scans and 360° images of 23 human heads with associated camera calibrations, and AFHQv2 Cats [Choi et al. 2020; Karras et al. 2021], a collection of high-resolution in-the-wild portraits of cats.

5.1 Comparisons

Baselines. We compare our methods against three state-of-the art methods for 3D aware-image generation from a single image: ROME [Khakhulin et al. 2022], HeadNeRF [Hong et al. 2022a], and EG3D-PTI, which combines an unconditional EG3D generator [Chan et al. 2022] and Pivotal Tuning Inversion (PTI) [Roich et al. 2021]. We

Fig. 5. Qualitative results displaying our model's reconstruction on the input view, and the learned geometry from the frontal view. The reconstructed geometry remains faithful to the input image. Credits to Devon Weller, Jamie, SupportPDX, Mary Sawatzky, map, Herzliya Conference, Helse Midt-Norge, Tom Munnecke, pter tr, UGA CAES/Extension, Rare Cancers Australia, Vladimir Agafonkin, Michael E. Macmillan, Nguyen Hung Vu.

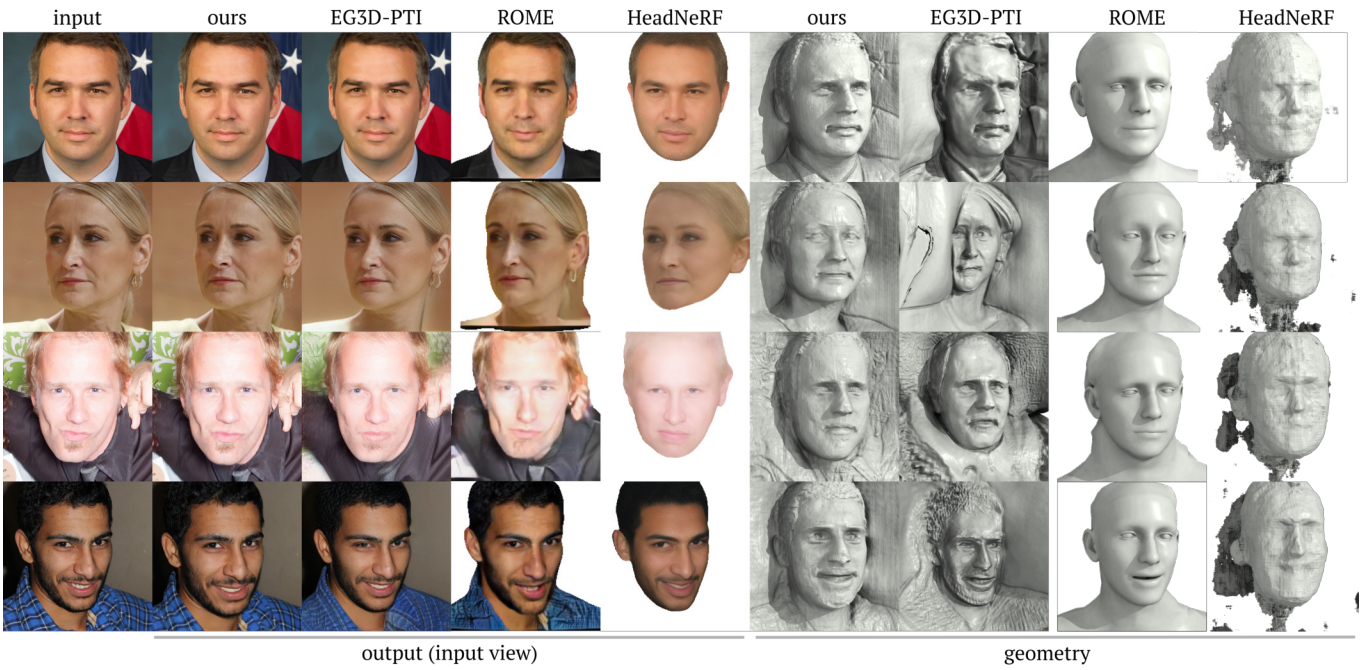

Fig. 6. Qualitative results displaying our model in comparison to baseline methods HeadNeRF, ROME, and EG3D-PTI, comparing the image quality (left) and reconstructed geometry (right). EG3D-PTI occasionally exhibits corrupted 3D geometry (2nd and 4th rows) when the input is side view, indicating that the learned 3D prior alone is not enough to ensure robust reconstruction. Credit to U.S. Dept. of HUD, Cristina Cifuentes, Rainforest Action Network, CENA MINEIRA.

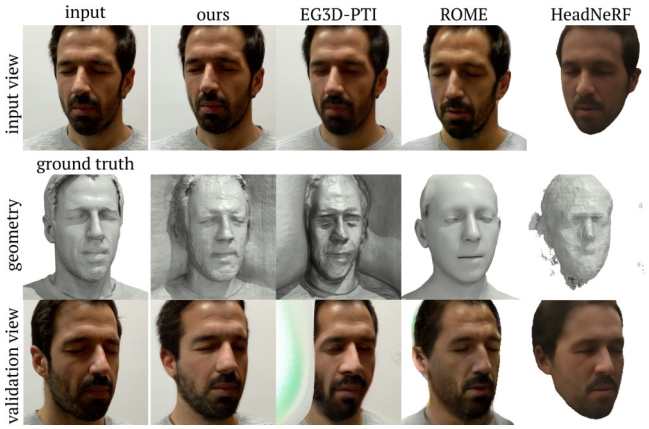

Fig. 7. Ground truth comparisons on the H3DS dataset including ground truth geometry (second row) and unseen validation view (third row). Since the H3DS ground truth data has inconsistent lighting, the lighting discrepancy is expected for the validation view.

also compare with EG3D itself as an unconditional reference on FID. We additionally provide extensive evaluations on our lightweight model (LT), which is introduced in Sec. 4.

Qualitative results. Fig. 1 shows our qualitative results on FFHQ and AFHQ. Fig. 4 and Fig. 5 show selected examples from FFHQ, demonstrating high-quality novel views and 3D geometry reconstructed by our method from a single portrait. Fig. 6 provides a qualitative comparison against baselines. While HeadNeRF and ROME provide adequate shapes and images, they need image segmentation as a preprocess, and struggle with obtaining photorealistic results. Despite the 20 mins of fine tuning, EG3D-PTI does not ensure the reconstruction looks photorealistic when viewed from a non-input view (see Fig. 2). In contrast, our method reconstructs the entire portrait with accurate photorealistic details. Fig. 7 provides comparisons to the ground truth validation view and 3D scan on H3DS. The synthesized image and 3D geometry of ROME and HeadNeRF generally lack the fidelity and reconstruct only a part of the head. EG3D-PTI occasionally outputs a degenerate 3D shape due to the highly unconstrained nature of single-view training of the NeRF representation (see Figs 1, 6 and 7). Our geometry retains overall the 3D shape as well as person-specific facial details. We also provide results on lifting 2D drawings and paintings into 3D in Fig. 8. While our method is never trained with stylized images, it can reasonably well handle those out-of-domain input images. Finally, we also show the outputs of our method in comparison to baselines at varying pitch and yaw in Figs 9 and 10, displaying the benefit of our method for photorealistic facial frontalization of challenging images. In comparison to baselines, our method's geometry does not collapse for challenging yaws as EG3D-PTI, and shows a significantly higher degree of photorealism than ROME and HeadNeRF.

Fig. 8. Qualitative results displaying our model's ability to lift StyleGAN2-generated drawings and paintings to 3D. These results display the generalizability of our model, as it can canonicalize out-of-domain drawings and portraits, lifting them to 3D.

Table 2. Quantitative evaluation using LPIPS, DISTS, SSIM, pose accuracy (Pose) and identity consistency (ID) on 500 FFHQ images. †Evaluated only using the foreground on 256^2 images. ‡Evaluated only using the face region.

	LPIPS↓	DISTS↓	SSIM↑	Pose↓	ID↑
HeadNeRF‡	.2502	.2427	.7514	.0644	.2031
Ours‡	**.1240**	**.0770**	**.8246**	**.0490**	**.5481**
ROME (256)†	.1158	.1058	.8257	.0637	.3231
Ours†	**.0468**	**.0407**	**.8981**	**.0486**	**.5410**
EG3D-PTI	.3236	.1277	**.6722**	.0575	.4650
Ours	**.2692**	**.0904**	.6598	.0485	**.5426**
Ours (LT)	.2750	.1021	.6655	**.0448**	.5404

Quantitative evaluations. Tab. 2 shows numerical comparisons of our method against baselines on 500 randomly selected images from FFHQ. We measure the 2D image reconstruction quality in the input view using LPIPS, DISTS, and SSIM. We evaluate multiview consistency using poses (Pose) estimated from synthesized images by an off-the-shelf pose detector [Deng et al. 2019b] following similar protocols as in previous work [Chan et al. 2022; Shi et al. 2021], and identity (ID) consistency by computing the mean of MagFace [Meng et al. 2021] (not used in our training) cosine similarity scores between the input view and synthesized view from a random camera pose. Since HeadNeRF and ROME only produce the face region and the foreground respectively, we also provide the same metrics from our models evaluated on the same masked region. Tab. 2 shows that our model significantly outperforms the baselines on all the metrics except SSIM; our SSIM score is only marginally lower than EG3D-PTI despite the aforementioned issue of the image misalignment and the fact that EG3D-PTI directly optimizes the pixels for the evaluation view. The geometry evaluation in Tab. 3 on H3DS in which we compare the depths of the ground truth from the input view as predicted by each model validates that our models produce more accurate 3D geometry.

Table 3. Scale- and translation-invariant depth evaluation using ground truth geometry from H3DS datasets. †Evaluated only using the face region.

Depth	H.NeRF	ROME	EG3D-PTI	Ours	Ours (LT)
L1↓	0.108†	0.054	0.071	**0.048**	0.049
RMSE↓	0.147†	0.084	0.101	**0.074**	0.075

5.2 Ablation study

We provide ablation studies comparing variants of our architecture and different training strategies. All variants are evaluated after training with 3M images.

Inference time and number of parameters. We compare the performance of two variants of our model, which have the same architecture but have different numbers of parameters and resolution of intermediate feature maps: "Ours" (87M params) and "Ours

Fig. 9. Comparison to baselines at various input pitch angles. Credits to Bjørnar Tollaksen, Juliana Martuscelli, The 621st Contingency Response Wing, U.S. Army Security Assistance Command, Sam Wadman, Laity Lodge Family Camp, U.S. President's Malaria Initiative, SickKids Foundation.

(LT)" (63M params). Tab. 1 provide runtime comparisons of the two. Tabs. 2 and 3 provide several comparisons of the two on image reconstruction, the accuracy of 3D shapes and identity consistencies. These extensive evaluations suggest that our lightweight model retains very close performance to our full model despite running significantly faster. Figs. 1 and 4 show qualitative samples from both our full model and our lightweight model.

Effects of Transformers. Fig. 11 compares results obtained with or without the proposed Transformer layers in the encoder. For this variant, we replaced the ViT module with CNN with matching number of parameters. Tab. 4 provides numerical comparisons of the two variants on image and 3D quality metrics. These quantitative and qualitative comparisons show that the ViT layers are important for creating more accurate 3D representations as well as achieving more accurate 2D image reconstruction.

Effects of camera augmentation. Fig. 12 compares the models trained with or without the camera augmentation for robustness to camera noise (also see the first row of Fig. 6 for the results of

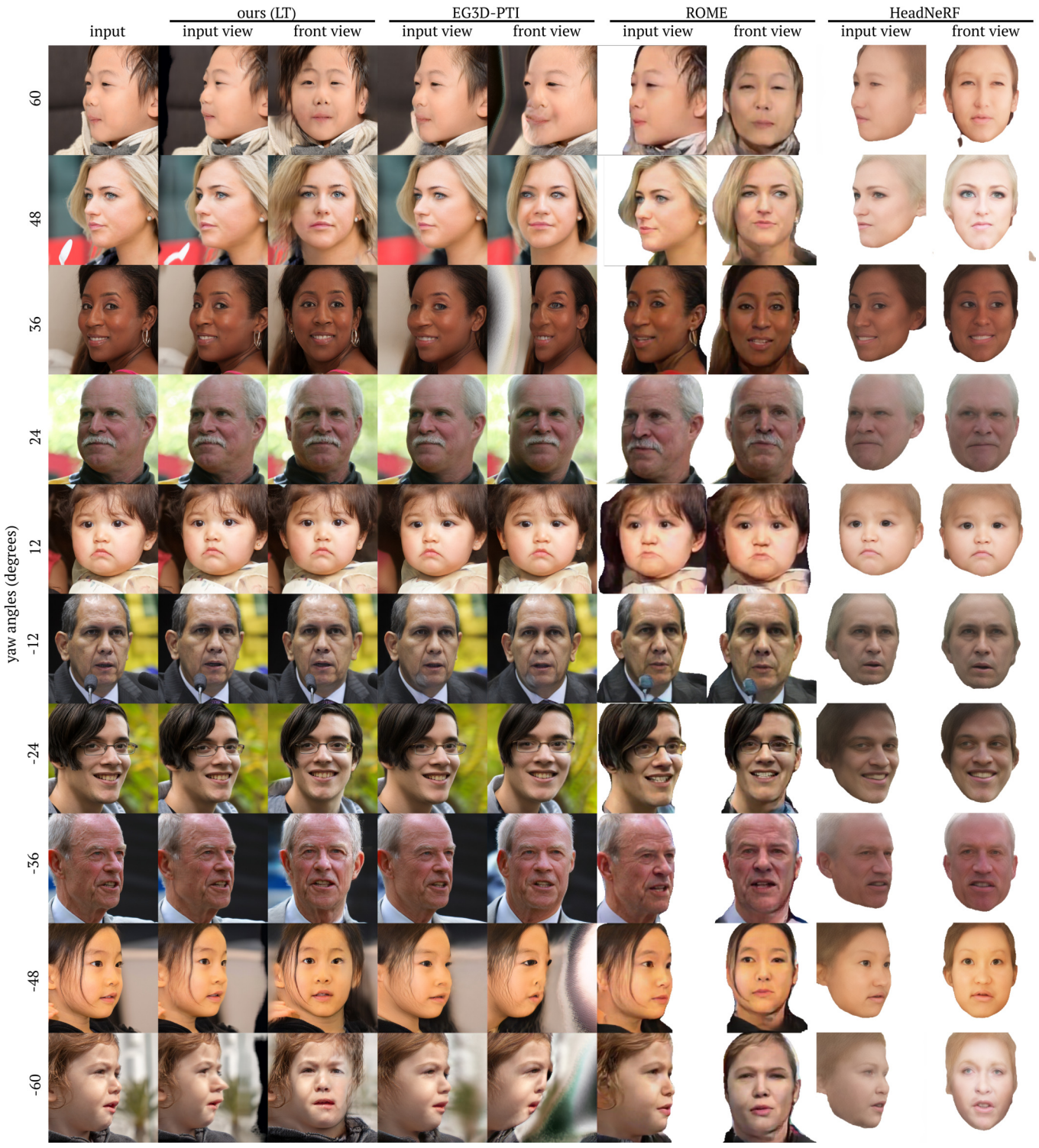

Fig. 10. Comparison to baselines at various input yaw angles. Credit to Lionel AZRIA, justinkim1, nonorganical, Paradox Wolf, Agência Senado, Ademir Brito, Ariana Vincent, Jay Weenig, John Benson, Seong Bae.

the same subject without the camera noise). We fix the camera calibration and apply image space rotation, translation, and zoom to the input image, emulating the effect of inaccurate camera extrinsics and intrinsics. Although our model does not rely on any camera information for canonicalization, the result is not robust without the proposed camera augmentation. EG3D-PTI assumes a fixed image alignment used to train the GAN model and is very sensitive to small image misalignment in the input. Tab. 4 provides numerical comparisons of our model with and without the proposed augmentation.

Effects of fine-tuned synthetic discriminator. We provide an additional ablation on the discriminator loss (L_{adv} in Eqn. 5), which fine-tunes the pre-trained EG3D discriminator with EG3D-generated images. As seen in Tab. 5, removing this discriminator loss results in a worse FID score. Moreover, as seen in Fig. 13, the renderings of our proposed method are significantly sharper with the synthetic discriminator tuning. Please see Sec. A1 and Tab. A2 for attempts to train the discriminator with real images.

Table 4. Ablation studies evaluating the proposed camera augmentation and the Transformer module. Without augmentation, the model acts as an autoencoder, mapping real images to arbitrary 3D representations that resemble the input (good ID score), but are not actually 3D (poor Pose score). Without a transformer, the encoder can canonicalize the inputs well (good Pose score), but cannot represent the details of the input (poor ID score). Our full method achieves both good Pose and ID scores with high reconstruction quality.

	LPIPS↓	DISTS↓	Pose↓	ID↑	FID↓
No aug.	0.3846	.1286	0.1758	**0.5359**	**3.42**
No Transformer	0.5419	.1650	**0.0426**	0.1906	11.5
Ours	**0.2894**	**.1053**	0.0461	0.5230	4.45

Table 5. Comparison in FID between our model and an ablated model without the synthetic discriminator.

FID↓	FFHQ
w/o synthetic disc.	7.71
Ours	**4.45**

5.3 Application: real-time 3D telepresence

We apply our method for lifting a monocular RGB video input to 3D in real-time, as would be needed for 3D telepresence. Our method processes the video frame by frame. Despite being trained on individual frames of synthetic data and processing the input video in a frame to frame fashion, our method can provide reasonable temporal consistency. Please refer to the teaser Fig. 1 (bottom right) for the output from our lightweight model as well as video examples from the supplement. Fig. 14 shows our system set up and running off of a desktop with a single RTX 4090. Our method can lift a monocular RGB video frame from a mobile phone to 3D in real-time.

Fig. 11. Ablation study comparing our model with and without the proposed Transformer modules. The model w/o Transformer replaces all Transformer Blocks with resolution-preserving residual CNNs with similar parameters. Credit to Kirill Chebotar.

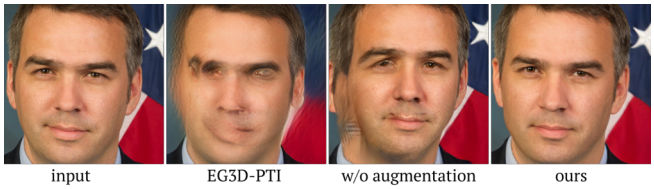

Fig. 12. Camera augmentation ablation study. Note that this is the same image as the first row of Fig. 6 except rotated and cropped non-centrally. Without augmentation, our result exhibits artifacts when the input image has zoom or camera roll. Similarly, EG3D-PTI is also sensitive to the image misalignment, as the camera pose becomes noisy, while our method correctly canonicalizes the face. Images are cropped and aligned for visual consistency. Credit to U.S. Dept. of HUD.

Fig. 13. Comparison between our model and an ablated model trained without the synthetic discriminator. Note the blurriness without the adversarial loss. Credit to Mohd Fazlin Mohd Effendy Ooi.

6 DISCUSSION

Limitation. When the input is a strong profile view (e.g., 60 degrees yaw angle), our method may struggle with properly canonicalizing the input, as it is highly out-of-distribution with respect to EG3D-generated images and FFHQ. Please see Figs. 9 and 10 for various challenging levels of pitch and yaw for input images. While our method can predict a canonicalized 3D representation without requiring camera poses as input, the rendered image may be slightly misaligned when compared to the input view (see Fig. A8 in the

Fig. 14. Our system applied to create a 3D telepresence live from a monocular RGB input. Please see the supplement video for the live demonstration.

supplement for the detailed analysis) possibly due to the combination of the imperfect canonicalization and noisy camera poses from an off-the-shelf pose estimator. Finally, although our method can provide reasonable temporal consistency when applied to a video in a frame-by-frame fashion, temporal inconsistencies remain as the canonicalizations change slightly per frame, and the predicted camera poses are entirely independent.

Future work. In the future, combining our method with camera pose optimization [Ko et al. 2023] may lead to more accurate 3D reconstruction and camera pose estimation. Additionally, jointly predicting the camera poses and triplanes in an autoregressive or recurrent context [Kalchbrenner et al. 2017; Shi et al. 2015; Srivastava et al. 2015] may result in more consistent frame-by-frame results. Next, it would be interesting to incorporate real images in the training as our preliminary attempts did not yield improvements. Finally, as our pipeline does not necessarily assume any category-specific priors, we can view it as a general method to distill the knowledge of a 3D GAN into a feedforward encoder. Thus, extending 3D GANs to more general scenes [Skorokhodov et al. 2023] may allow our pipeline to create 3D representations of arbitrary scenes in the future. Specifically extending our work to handle hands or the full body, is of interest for real-time telepresence applications.

Conclusion. We proposed a one-shot encoder-based framework to lift a single RGB image to 3D in real-time and demonstrated our method, trained entirely from synthetic data, can handle challenging (even out-of-domain) real-world images. We believe that this opens up possibilities for accessible 3D reconstructions of real-world objects and interactive 3D visualization from a picture.

ACKNOWLEDGEMENTS

We thank David Luebke, Jan Kautz, Peter Shirley, Alex Evans, Towaki Takikawa, Ekta Prashnani and Aaron Lefohn for feedback on drafts and early discussions. We acknowledge the significant efforts and suggestions of the reviewers. For allowing the use of video, we thank Elys Muda. This work was funded in part at UCSD by ONR grants N000142012529, N000142312526, an NSF graduate Fellowship, a Jacobs Fellowship, and the Ronald L. Graham chair. Manmohan Chandraker acknowledges support of NSF IIS 2110409. Koki Nagano and Eric Chan were partially supported by DARPA's Semantic Forensics (SemaFor) contract (HR0011-20-3-0005). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?. In *IEEE International Conference on Computer Vision (ICCV)*.

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. 2021. ReStyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. In *IEEE International Conference on Computer Vision (ICCV)*.

ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. 2022. RigNeRF: Fully Controllable Neural 3D Portraits. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ang Cao, Chris Rockwell, and Justin Johnson. 2022. FWD: Real-time Novel View Synthesis with Forward Warping and Depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. 2022. Efficient Geometry-aware 3D Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. 2021. pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *IEEE International Conference on Computer Vision (ICCV)*.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

S Chen and L Williams. 1993. View Interpolation for Image Synthesis. In *SIGGRAPH 93*. 279–288.

Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. 2019a. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. 2022. GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. 2019b. Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set. In *IEEE Computer Vision and Pattern Recognition Workshops*.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. 2022. Image Quality Assessment: Unifying Structure and Texture Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. 2022. HyperInverter: Improving StyleGAN Inversion via Hypernetwork. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. 2021. HeadGAN: One-shot Neural Head Synthesis and Editing. In *IEEE International Conference on Computer Vision (ICCV)*.

Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. 2022. Megaportraits: One-shot megapixel neural head avatars. *arXiv preprint arXiv:2207.07621* (2022).

Qianli Feng, Viraj Shah, Raghudeep Gadde, Pietro Perona, and Aleix Martinez. 2022. Near perfect gan inversion. *arXiv preprint arXiv:2202.11833* (2022).

Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. 2020. Portrait Neural Radiance Fields from a Single Image. *arXiv preprint arXiv:2012.05903* (2020).

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*.

S Gortler, R Grzeszczuk, R Szeliski, and M Cohen. 1996. The Lumigraph. In *SIGGRAPH 96*. 43–54.

Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 2018. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. 2021. StyleNeRF: A Style-based 3D-Aware Generator for High-resolution Image Synthesis. *arXiv preprint arXiv:2110.08985* (2021).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. 2022b. Depth-Aware Generative Adversarial Network for Talking Head Video Generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. 2022a. HeadNeRF: A Real-time NeRF-based Parametric Head Model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wonbong Jang and Lourdes Agapito. 2021. Codenerf: Disentangled neural radiance fields for object categories. In *IEEE International Conference on Computer Vision (ICCV)*.

N. Khademi Kalantari, T. Wang, and R. Ramamoorthi. 2016. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (SIGGRAPH Asia 16)* 35, 6 (2016), 193:1–193:10.

Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. 2017. Video pixel networks. In *International Conference on Machine Learning*. PMLR, 1771–1779.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. 2022. Realistic One-shot Mesh-based Head Avatars. In *European Conference on Computer Vision (ECCV)*.

Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. 2018. Deep Video Portraits. *ACM Transactions on Graphics (SIGGRAPH)* (2018).

Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 2023. 3D GAN Inversion with Pose Optimization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Thomas Leimkühler and George Drettakis. 2021. Freestylegan: Free-view editable portrait rendering with the camera manifold. *arXiv preprint arXiv:2109.09378* (2021).

M Levoy and P Hanrahan. 1996. Light Field Rendering. In *SIGGRAPH 96*. 31–42.

Xingyi Li, Chaoyi Hong, Yiran Wang, Zhiguo Cao, Ke Xian, and Guosheng Lin. 2022. SymmNeRF: Learning to Explore Symmetry Prior for Single-View View Synthesis. In *Asian Conference on Computer Vision (ACCV)*.

C.Z. Lin, D.B. Lindell, E.R. Chan, and G. Wetzstein. 2022. 3D GAN Inversion for Controllable Portrait Image Animation. In *ECCV Workshop on Learning to Generate 3D Shapes and Scenes*.

Kai-En Lin, Lin Yen-Chen, Wei-Sheng Lai, Tsung-Yi Lin, Yi-Chang Shih, and Ravi Ramamoorthi. 2023. Vision Transformer for NeRF-Based View Synthesis from a Single Input Image. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

N. Max. 1995. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (1995).

L McMillan and G Bishop. 1995. Plenoptic Modeling: An Image-Based Rendering System. In *SIGGRAPH 95*. 39–46.

Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. 2021. MagFace: A universal representation for face recognition and quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. 2022. KeypointNeRF: Generalizing Image-based Volumetric Avatars using Relative Spatial Encoding of Keypoints. In *European Conference on Computer Vision (ECCV)*.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*.

Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. 2019. Deep face normalization. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.

Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. 2018. PaGAN: Real-Time Avatars Using Dynamic Textures. *ACM Transactions on Graphics (SIGGRAPH ASIA)* (2018).

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. 2019. HoloGAN: Unsupervised learning of 3D representations from natural images. In *IEEE International Conference on Computer Vision (ICCV)*.

Michael Niemeyer and Andreas Geiger. 2021. GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. 2022. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. 2021. Do 2D GANs Know 3D Shape? Unsupervised 3D Shape Reconstruction from 2D Image GANs. In *International Conference on Learning Representations (ICLR)*.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei Efros, and Eli Shechtman. 2022. GAN-Supervised Dense Visual Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. 2021. H3D-Net: Few-Shot High-Fidelity 3D Head Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*.

Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. 2022. LOLNeRF: Learn from One Look. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xuanchi Ren and Xiaolong Wang. 2022. Look Outside the Room: Synthesizing A Consistent Long-Term 3D Scene Video from A Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. *arXiv preprint arXiv:2106.05744* (2021).

R. Rombach, P. Esser, and B. Ommer. 2021. Geometry-Free View Synthesis: Transformers and no 3D Priors. In *IEEE International Conference on Computer Vision (ICCV)*.

Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. 2022. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. 2020. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yeongho Seol, Jaewoo Seo, Paul Hyunjin Kim, J. P. Lewis, and Junyong Noh. 2011. Artist Friendly Facial Animation Retargeting. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yichun Shi, Divyansh Aggarwal, and Anil K Jain. 2021. Lifting 2D StyleGAN for 3D-Aware Face Generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 2023. 3D generation on ImageNet. *arXiv preprint arXiv:2303.01416* (2023).

Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. 2022. EpiGRAF: Rethinking training of 3D GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. 2017. Learning to Synthesize a 4D RGBD Light Field from a Single Image. In *International Conference on Computer Vision (ICCV)*. 2262–2270.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*. PMLR, 843–852.

Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. 2022. IDE-3D: Interactive Disentangled Editing for High-Resolution 3D-aware Portrait

Synthesis. *ACM Transactions on Graphics (SIGGRAPH ASIA)* (2022).
Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, and Ravi Ramamoorthi. 2021. NeLF: Neural Light-transport Field for Portrait View Synthesis and Relighting. In *Eurographics Symposium on Rendering*.
Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an Encoder for StyleGAN Image Manipulation. *ACM Transactions on Graphics (SIGGRAPH)* (2021).
Alex Trevithick and Bo Yang. 2021. GRF: Learning a General Radiance Field for 3D Scene Representation and Rendering. In *IEEE International Conference on Computer Vision (ICCV)*.
Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. 2021. Repurposing GANs for One-shot Semantic Part Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Daoye Wang, Prashanth Chandran, Gaspard Zoss, Derek Bradley, and Paulo Gotardo. 2022a. MoRF: Morphable Radiance Fields for Multiview Neural Head Modeling. In *ACM SIGGRAPH 2022 Conference Proceedings*.
Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. 2021b. IBRNet: Learning Multi-View Image-Based Rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. 2022c. High-Fidelity GAN Inversion for Image Attribute Editing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 2021a. One-Shot Free-View Neural Talking-Head Synthesis for Video Conferencing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. 2022b. Latent Image Animator: Learning to Animate Images via Latent Space Navigation. In *International Conference on Learning Representations (ICLR)*.
Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *TIP* (2004).
Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. SynSin: End-to-end View Synthesis from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljevic, Daniel Wilde, Stephan Garbin, Chirag Raman, Jamie Shotton, Toby Sharp, Ivan Stojiljkovic, Tom Cashman, and Julien Valentin. 2022. 3D face reconstruction with dense landmarks. In *European Conference on Computer Vision (ECCV)*.
Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. 2021. Fake It Till You Make It: Face Analysis in the Wild Using Synthetic Data Alone. In *IEEE International Conference on Computer Vision (ICCV)*.
Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. 2022. Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds. *arXiv preprint arXiv:2206.07255* (2022).
Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.
Jiaxin Xie, Hao Ouyang, Jingtan Piao, Chenyang Lei, and Qifeng Chen. 2022a. High-fidelity 3D GAN Inversion by Pseudo-multi-view Optimization. *arXiv preprint arXiv:2211.15662* (2022).
Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2022b. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library.
Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Humphrey Shi, and Zhangyang Wang. 2022a. SinNeRF: Training Neural Radiance Fields on Complex Scenes from a Single Image. In *European Conference on Computer Vision (ECCV)*.
Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 2022b. 3D-aware Image Synthesis via Learning Structural and Textural Representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Yu-Ying Yeh, Koki Nagano, Sameh Khamis, Jan Kautz, Ming-Yu Liu, and Ting-Chun Wang. 2022. Learning to Relight Portrait Images via a Virtual Light Stage and Synthetic-to-Real Adaptation. *ACM Transactions on Graphics (SIGGRAPH ASIA)* (2022).
Fei Yin, Yong Zhang, Xuan Wang, Tengfei Wang, Xiaoyu Li, Yuan Gong, Yanbo Fan, Xiaodong Cun, Öztireli Cengiz, and Yujiu Yang. 2022. 3D GAN Inversion with Facial Symmetry Prior. *arxiv:2211.16927* (2022).
Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Xianggang Yu, Jiapeng Tang, Yipeng Qin, Chenghong Li, Xiaoguang Han, Linchao Bao, and Shuguang Cui. 2022. PVSeRF: Joint Pixel-, Voxel- and Surface-Aligned Radiance Field for Single-Image Novel View Synthesis. In *ACM International Conference on Multimedia*.
Egor Zakharov, Aleksei Ivakhnenko, Aliaksandra Shysheya, and Victor Lempitsky. 2020. Fast Bi-layer Neural Synthesis of One-Shot Realistic Head Avatars. In *European Conference on Computer Vision (ECCV)*.
Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Xuaner Zhang, Jonathan T. Barron, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, and David E. Jacobs. 2020. Portrait Shadow Manipulation. *ACM Transactions on Graphics (SIGGRAPH)*.
Xuanmeng Zhang, Zhedong Zheng, Daiheng Gao, Bang Zhang, Pan Pan, and Yi Yang. 2022. Multi-View Consistent Generative Adversarial Networks for 3D-aware Image Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. 2021. DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Jian Zhao and Hui Zhang. 2022. Thin-plate spline motion model for image animation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3657–3666.
Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. 2021. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. *arXiv preprint arXiv:2110.09788* (2021).

Supplementary Material
Real-Time Radiance Fields for Single-Image Portrait View Synthesis

ALEX TREVITHICK*, University of California San Diego, USA
MATTHEW CHAN and MICHAEL STENGEL, NVIDIA, USA
ERIC R. CHAN*, Stanford University, USA
CHAO LIU, ZHIDING YU, and SAMEH KHAMIS, NVIDIA, USA
MANMOHAN CHANDRAKER and RAVI RAMAMOORTHI, University of California San Diego, USA
KOKI NAGANO, NVIDIA, USA

ACM Reference Format:
Alex Trevithick, Matthew Chan, Michael Stengel, Eric R. Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano. 2023. Supplementary Material Real-Time Radiance Fields for Single-Image Portrait View Synthesis. *ACM Trans. Graph.* 1, 1, Article 1 (January 2023), 10 pages. https://doi.org/10.1145/3592460

In this supplement, we first provide the additional results including additional evaluations and comparisons in Sec. A1. We provide the implementation details of our models, including architecture details, camera augmentation, training details, and hyper parameters in Sec. A2. We also provide further experiment details in Sec. A3. Finally, we discuss the limitations of our work in Sec. A4. We encourage the readers to view our accompanying videos in the supplement, which include the additional visual comparisons, results, and live demonstration of the novel view synthesis from a video input.

A1 ADDITIONAL RESULTS

A1.1 Additional qualitative results

We provide additional qualitative results generated from a single input image from FFHQ in Fig. A1 and AFHQ in Fig. A2. Fig. A1 shows that our method can handle complex hairstyles (first row), and asymmetric facial expressions (second and third rows). Fig. A2 shows our method can handle unconstrained poses of cats present in the portraits as well as a wide variety of their textures.

*This project was initiated and substantially carried out during an internship at NVIDIA.

Authors' addresses: Alex Trevithick, University of California San Diego, La Jolla, USA; Matthew Chan; Michael Stengel, NVIDIA, Santa Clara, USA; Eric R. Chan, Stanford University, Stanford, USA; Chao Liu; Zhiding Yu; Sameh Khamis, NVIDIA, Santa Clara, USA; Manmohan Chandraker; Ravi Ramamoorthi, University of California San Diego, La Jolla, USA; Koki Nagano, NVIDIA, Santa Clara, USA.

0730-0301/2023/1-ART1
https://doi.org/10.1145/3592460

A1.2 Qualitative comparisons to [Ko et al. 2023]

In Fig. A3, we provide comparisons to the state-of-the-art 3D GAN inversion work by [Ko et al. 2023]. While their method needs test-time optimization for the camera parameters and generator tuning, our method can process an unposed input in one-shot.

A1.3 Additional comparisons

We provide additional comparisons to HeadNeRF, ROME, and EG3D-PTI in Fig. A6. HeadNeRF only reconstructs the head region and struggles to reconstruct out-of-domain hair color (first row). ROME reconstructs the foreground image well, but requires background segmentation and the geometry does not fully capture the hairstyles and eyeglasses (second and third rows). EG3D-PTI reconstructs full RGB images and geometry, but occasionally produces distorted 3D shapes (first row, better viewed in 3D in the accompanying video) when the input view is non-frontal. Our method produces consistent image and geometry reconstruction quality across the variety of inputs including a non-realistic human image (fourth row).

A1.4 Percentile results based on LPIPS

In Fig. A7, we show our results on FFHQ and AFHQ shown in the order of the LPIPS percentile scores. For FFHQ, we use the same randomly selected 500 FFHQ test set described in the main manuscript and for AFHQ, we randomly selected 485 images for which we computed the LPIPS scores. The percentile results preferred by the LPIPS scores show that our method can demonstrate consistent quality for the large portion of the test images.

PSNR and SSIM on misaligned images. We provide our analysis on PSNR and SSIM metrics on images when images are aligned and when images have a small misalignment in Fig. A8. While LPIPS scores can tolerate a small image misalignment (little change when images are aligned or misaligned), the PSNR and SSIM scores significantly change, which make these metrics unreliable for our tasks when the reconstructed images are not perfectly pixel-to-pixel aligned. The issues of PSNR and SSIM scores sensitivity under geometry transformation are reported by previous work [Ding et al. 2022]. The DISTS [Ding et al. 2022] metric can also tolerate slight misalignment.

A1.5 Evaluation of FID

Tab. A1 provides comparisons on FID calculated over 50K images from FFHQ and 10K images from AFHQ. Our lightweight model

Fig. A1. Additional qualitative results generated by our method on FFHQ. Credits to USAID | Southern Africa, TimothyJ, toan đào song, Travis Rock, Curt Mills, UGA CAES/Extension.

Fig. A2. Additional qualitative results generated by our method on AFHQ.

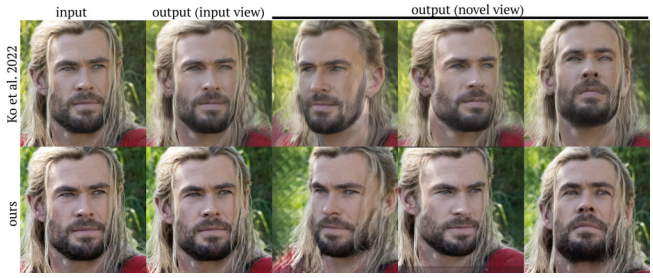

Fig. A3. Qualitative comparisons to the concurrent work[Ko et al. 2023] that relies on test-time camera optimization and generator weights tuning.

Fig. A4. Visualization of the limits in pitch and yaw of the camera pose distribution for synthetic input images used to supervise our model.

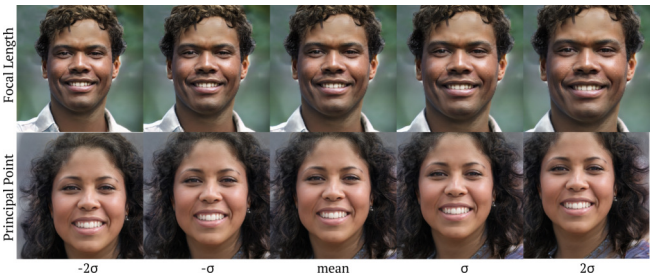

Fig. A5. Visualization of two sigmas of noise in the principal point and focal length used for camera augmentation during our training.

Table A1. Comparisons to an *unconditional* reference on FID evaluated over 50K images of FFHQ and 10k images of AFHQ (including horizontal flips). †Using transfer learning from a pretrained FFHQ model.

FID↓	FFHQ	AFHQ
EG3D	4.05	2.88†
Ours	**3.48**	2.39†
Ours (LT)	4.25	**2.11**†

"Ours (LT)" produces competitive FID scores to our full model ("Ours").

Table A2. Additional ablation study comparing variants of our models. "10K" refers to when we pre-compute 10K triplanes (subjects) and generate supervising views on the fly using EG3D. "w/real data" shows preliminary results of our initial attempt to incorporate real images in the training.

	LPIPS↓	DISTS↓	Pose↓	ID↑	FID↓
10K	**0.2797**	**0.995**	**0.0458**	0.5000	4.60
w/real data	0.3060	0.1125	0.0539	0.4556	6.15
Ours	0.2894	.1053	0.0461	**0.5230**	**4.45**

A1.6 Ablation study

We provide additional ablation studies concerning the importance of the training dataset size and describe our preliminary attempt to incorporate real images into the training.

Adding real data to the training. We attempted to incorporate real data into the training pipeline in a variety of ways, but each one proved unsuccessful. Our most succesful attempt was to train the real part of the discriminator with images from FFHQ (using the same conditioning as the original EG3D) and add significant noise to the discriminator pose conditioning. Tab. A2 shows the results of our preliminary attempt to incorporate real images in the training. As can be seen in Fig. A9, even this method fails to reconstruct the input image faithfully.

Size of training data. We additionally performed an ablation on the number of subjects in the training set. To do so, we chose 10k latent codes from EG3D and rendered images from only these. We found that training with 10k subjects with on the fly supervising view generation (theoretically each subject has infinite views to supervise) performs similarly to our method which synthesizes new identities on the fly, as seen in Tab. A2. We hypothesize that this is because the number of subjects is similar to the datasets, such as VGGFace2 [Cao et al. 2018] (9K subjects) and CASIA-WebFace [Yi et al. 2014] (10K subjects), used to train a one-shot face recognition model.

A1.7 Additional Applications

Portrait frontalization. Our method can be applied to portrait face frontalization, which is useful for 3D reconstruction and avatar digitization [Nagano et al. 2019]. Please see the examples in Figs. A1 (4th column), A2 (4th column), A7 (3rd and 7th columns).

A2 IMPLEMENTATION DETAILS

We implement our framework in PyTorch on top of the official EG3D codebase (https://github.com/NVlabs/eg3d).

EG3D pre-trained model. For human faces, we use the EG3D model trained on the FFHQ dataset (*ffhqrebalanced512-128.pkl*). To simplify our encoder training supervision, we replaced the latent code W injected to the StyleGAN2-based super-resolution layer with a constant 1 and fine tuned the entire EG3D models on FFHQ for additional 6.8 M images of training. This resulted in the FID score 4.05 for FFHQ as reported in the main manuscript. For cats faces, we performed transfer learning [Karras et al. 2020] from this FFHQ checkpoint and trained additional 3.2M images on the cat split of

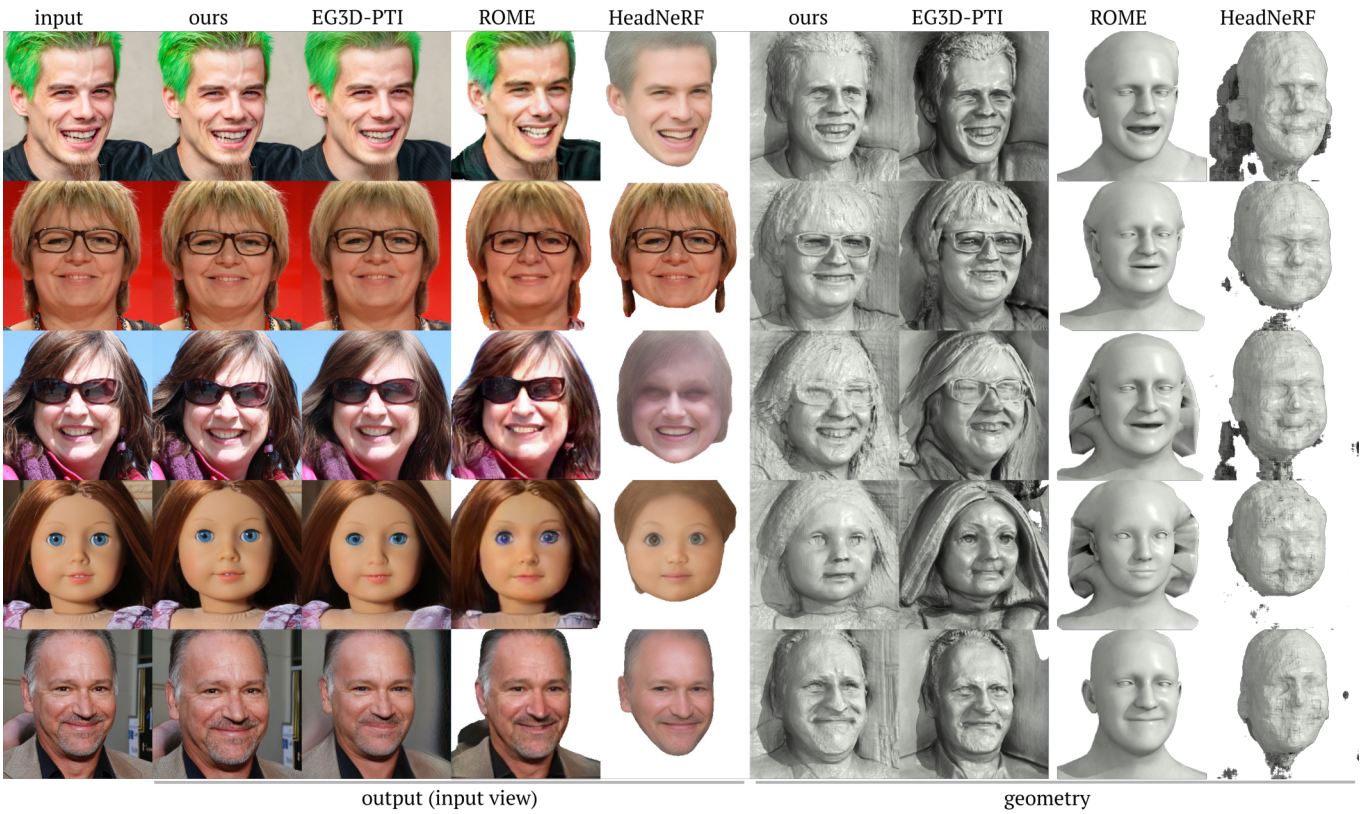

Fig. A6. Additional qualitative comparisons against baselines on input view reconstruction and geometry. Credit to Steffen Geyer, Force Ouvrière, Matt Hamm, scarlett1854, The Society of Motion Picture and Television Engineers.

AFHQv2, which resulted in the FID score 2.88, again reported in the main manuscript. Please refer to the samples of synthetic data generated by the EG3D model in Figs. A4 and A5.

Encoder for $\boldsymbol{F}_{low}$. We modify the first layer of the DeepLabV3 [Chen et al. 2017] architecture from the Pytorch Segmentation Models repo [Iakubovskii 2019] by concatenating the 2D pixel coordinate of each pixel, so that the input is 5 channels. We also remove all instances of batch norm (reintroducing the biases in all of the convolutional layers). Otherwise, we use the standard encoder-decoder as implemented with a ResNet34 encoder. We take the feature map output of the decoder of DeepLabV3 (the layer before bilinear upsampling and segmentation head). As seen in the top half of the pipeline figure in the main paper, this gives us a feature map $\boldsymbol{F}_{\text{low}}$.

Encoder for $\boldsymbol{F}$ with Conv Layers. $\boldsymbol{F}_{\text{low}}$ is fed to a hybrid convolutional-transformer architecture. We will denote OverLapPatchEmbed as the patchwise embedding from Segformer [Xie et al. 2021] with patch_size=3, and TransformerBlock as the efficient self-attention block from Segformer [Xie et al. 2021] without dropout, with kqv_bias, and with layer normalization. Then the DeepLabV3 decoder features are fed to the module given in Fig. A10, which outputs the low-resolution canonical features $\boldsymbol{F}$ as seen in the top half of the main paper's pipeline figure. Note that the output of this module is not technically $\boldsymbol{F}$, and instead $\boldsymbol{F}$ after being processed by additional convolutional layers.

Encoder for $\boldsymbol{F}_{high}$. We then encode the image again (with its stacked pixel coordinates) with $\mathbf{E}_{\text{high}}$ with architectures given in A11. Note that the input to the LT model's high-resolution encoder is the second layer output features of DeepLabV3, rather than the raw conditioning image.

Final triplane encoder. Finally, $\boldsymbol{F}$ and $\boldsymbol{F}_{\text{high}}$ are concatenated and decoded to the triplane $\boldsymbol{T}$ with the architectures seen in Fig. A12, completing the final encoding stage seen in the main paper's pipeline figure.

Misc. For super-resolution, we used the same super-resolution network architecture as EG3D, but replaced the w to be constant 1 as mentioned earlier. For volume rendering and decoding the triplane, we follow EG3D; specifically, we use 48 depth samples for coarse and fine passes for training. For discriminator, we use 2D dual-disciriminator from EG3D.

Training. In practice, we alternate between taking gradient steps with reference view supervision and multiview supervision. To do so, we begin by synthesizing synthetic input images for our encoder by sampling from the distribution for $\boldsymbol{P}_{\text{ref}}$ as detailed in the main paper. We render these cameras from triplanes from the frozen, pretrained

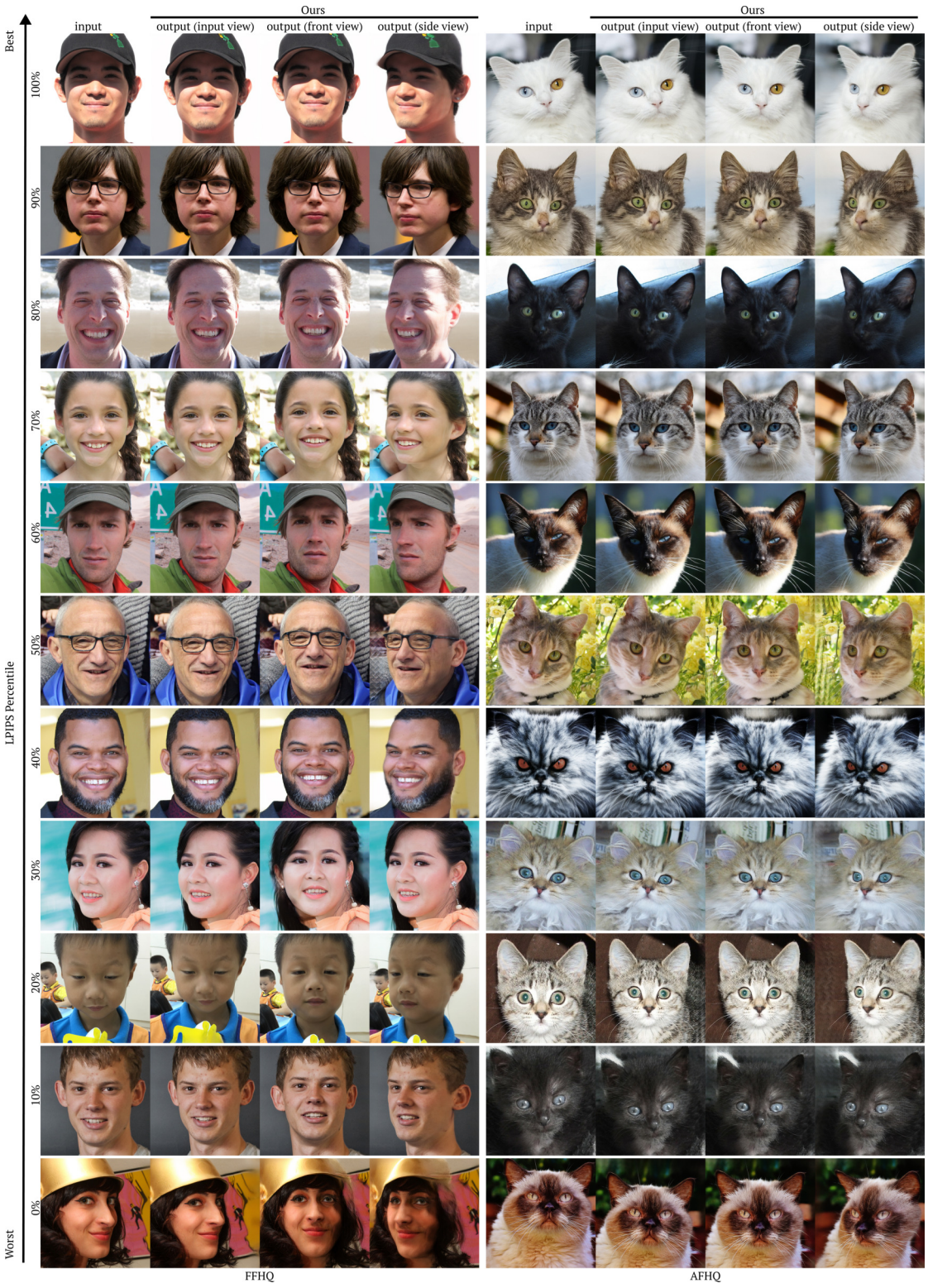

Fig. A7. Results generated by our method on FFHQ and AFHQ shown in the order of percentile. Note that percentiles for FFHQ are calculated with alignment, whereas AFHQ percentiles are calculated without alignment from a test set. Credits to yasminehabib, Rutgers Council on Public and Internation Affairs, davitydave, Laity Lodge Family Camp, Houston Marsh, Ordiziako Jakintza Ikastola, Edgar Caraballo, NGÁO STUDIO, Debbie, WorldSkills UK, Craig Duffy.

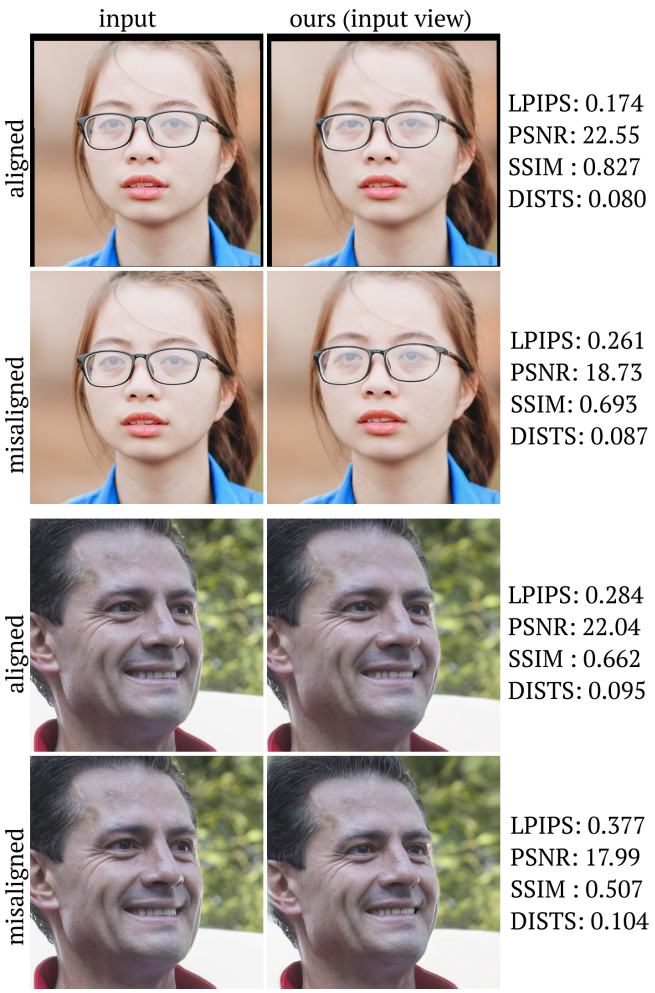

Fig. A8. LPIPS, DISTS, PSNR, and SSIM scores computed on images that have a small misalignment. Previous work [Ding et al. 2022] reported that LPIPS can tolerate small geometric misalignment better than PSNR and SSIM; DISTS is most robust to small image misalignment. Credits to Dong Quang, Presidencia de la Republica Mexicana.

Fig. A9. Comparisons showing the output of our initial attempt in incorporating real images. Credit to Mario Krajčír.

EG3D. These are then fed to our encoder, whereby a triplane is then predicted. We can then render the same input cameras to take a gradient step for a loss computed only over the input views. We can additionally sample some cameras P_{mv}, render ground truth from the EG3D triplanes, and render from the predicted triplanes for a multiview loss as well. In particular, at every gradient step, we always render 32 input cameras and 32 multiview cameras from the aforementioned distributions from the frozen EG3D. However, we do not always perform a gradient step for the input view loss.

In the first stage of the training, we compute losses for the reference set of cameras once every 10 triplane syntheses and perform gradient steps with respect to multiview supervision at every gradient step. We additionally do not incorporate any adversarial loss, nor category loss, and do not train the MLP decoder and super-resolution network parameters at all. We train for 30k iterations without these objectives in this first stage. In the second stage, we add the adversarial and category losses and backpropagate to all parameters in the pipeline, computing losses for the reference cameras every 2 gradient passes. In this stage, we remove the feature loss, and set the weight of the triplane loss to 0.01. After 37.5k iterations, we begin to compute multiview supervision and reference view supervision at every EG3D triplane synthesis step and continued to reach 220k iters in total (including the first 37.5k). We use a learning rate of 1e-4 for the encoder parameters, except for the transformer parameters, which have a learning rate of 5e-5. We use the same settings as EG3D for the the discriminator. We train for about 10 days on 8 A100 GPUs or 8 A40 GPUs, for about 220k iterations in total.

For training our model for cat faces, we used transfer learning following [Chan et al. 2022; Karras et al. 2020]. We initialized our cat face model with our human face model that is already trained, and ran training for additional 5 million images using the AFHQv2 checkpoint from EG3D.

Camera augmentation. EG3D assumes a fixed camera radius of 2.7, focal length of 18.83, zero camera roll, and a central principal point. For the FFHQ experiments, we sample the focal length from a normal distribution with standard deviation 1 centered at 18.83, the camera radius from a normal distribution centered at 2.7 with standard deviation 0.1, the principal point from a normal distribution with standard deviation 14 and centered at 256, and camera roll with a normal distribution of mean 0 and standard deviation 2 degrees.

For the AFHQ experiments, we sample the focal length from a normal distribution with standard deviation 1.5 centered at 18.83, the camera radius from a normal distribution centered at 2.7 with standard deviation 0.1, the principal point from a normal distribution with standard deviation 25 and centered at 256, and camera roll with a normal distribution of mean 0 and standard deviation 6 degrees.

Training data. We visualize the distribution of synthetic training data in two figures. Fig. A4 visualizes the limits of the input image poses in pitch and yaw for two subjects. Fig. A5 visualizes two sigmas of noise in the focal length and in the principal point used to augment camera information during our training.

Inference. To calculate the timings of our method, we wrap the forward calls of the encoder (not the rendering) in autocast, which

Encoder for F with Conv layers

Ours

```
OverlapPatchEmbed( img_size=64, stride=2, in_chans=256, embed_dim=1024 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
PixelShuffle(upscale_factor=2)
UpsamplingBilinear2d(scale =2.0, mode=bilinear)
Conv2d(256, 128, kernel =(3, 3), stride=(1, 1), padding=(1, 1))
ReLU()
UpsamplingBilinear2d(scale =2.0, mode=bilinear)
Conv2d(128, 128, kernel =(3, 3), stride=(1, 1), padding=(1, 1))
ReLU()
Conv2d(128, 96, kernel =(3, 3), stride=(1, 1), padding=(1, 1))
```

Ours (LT)

```
OverlapPatchEmbed( img_size=64, stride=2, in_chans=256, embed_dim=1024 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
TransformerBlock( dim=1024,num_heads=4, mlp_ratio=2, sr_ratio=1 )
PixelShuffle(upscale_factor=2)
UpsamplingBilinear2d(scale_factor=2.0, mode=bilinear)
Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Fig. A10. Details of the hybrid convolutional-transformer architecture which decodes the DeepLabV3 features before being concatenated with the high-resolution image features later on.

Encoder E_{high} for F_{high}

Ours

```
Conv2d(5, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3))
LeakyReLU(negative_slope=0.01)
Conv2d(64, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
```

Ours (LT)

```
Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Fig. A11. Details of E_{high} which maps the input image to a high-resolution feature map.

Final Encoder for Triplane

Ours

```
Conv2d(192, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
OverlapPatchEmbed( img_size=256, stride=2, in_chans=128, embed_dim=1024 )
TransformerBlock( dim=1024,num_heads=2, mlp_ratio=2, sr_ratio=2 )
PixelShuffle(upscale_factor=2)
Concat with output of Encoder for F with Conv Layers
Conv2d(352, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Ours (LT)

```
Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
OverlapPatchEmbed( img_size=128, stride=2, in_chans=128, embed_dim=1024 )
TransformerBlock( dim=256,num_heads=2, mlp_ratio=2, sr_ratio=2 )
PixelShuffle(upscale_factor=2)
Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
UpsamplingBilinear2d(scale_factor=2.0, mode=bilinear)
Conv2d(256, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(128, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
LeakyReLU(negative_slope=0.01)
Conv2d(96, 96, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Fig. A12. Details of the hybrid convolutional-transformer architecture which decodes the concatenated transformer features and high-resolution image features directly into a triplane representation.

we use for real-time applications. For renderings, we use 48 depth samples for real-time applications and 96 depth samples for the offline videos, following EG3D.

A3 EXPERIMENT DETAILS

A3.1 Baselines

For all the baselines we used, we used official code from the authors with released pre-trained checkpoints.

For HeadNeRF, we used the highest resolution model `model_Reso64` on the official website (https://github.com/CrisHY1995/headnerf), which produces the final output at 512 resolution using a feature map of resolution 64.

For ROME, we use the pre-trained model from the official code release (https://github.com/SamsungLabs/rome), which produces the output at 256 resolution.

For EG3D models, we used the FFHQ and AFHQv2 fine tuned models as described in Sec. A2, which are derived from the official EG3D models. The baseline "EG3D-PTI" combines the unconditional EG3D model with the lightweight generator tuning at test time using Pivotal Tuning Inversion (PTI) [Roich et al. 2021] for 3D GAN inversion from a single image. For the PTI inversion experiment, we follow the hyperparameter settings from the original PTI paper and the PTI experiment done in the EG3D paper, and optimize the latent code for 600 iterations, followed by fine tuning the generator weights for an additional 350 iterations. Unless noted otherwise, we used this setting for all our experiments.

FFHQ. For the comparisons on FFHQ between our model and other baselines, we postprocess our images with a rigid 2D warp. To accomplish this, we estimate 2D landmarks with an off-the-shelf facial landmark model [Bulat and Tzimiropoulos 2017] for both the ground truth and our predicted image. We then solve the Orthogonal Procrustes problem [Virtanen et al. 2020] to find the optimal orthogonal matrix to rigidly align our image onto the target image approximately around the face region using the facial landmarks. Examples of this alignment can be seen in Fig. A8. We found that this alignment resulted in worse performance for EG3D-PTI and HeadNeRF, so we do not postprocess these methods' renderings. For comparison to ROME, we align our renderings and ROME's renderings to ROME's warped input (lower-resolution) image with the same process before computing the metrics. In any cases where the warp produced black pixels on the border (out of bounds), we set the ground truth, and the baselines pixels to black there as well, to ensure that we are comparing the exact same pixels between the methods. For ROME and HeadNeRF, we compare only on their valid pixels, using the provided masks from these methods.

To ensure fairness in the ID and Pose comparisons of our model against HeadNeRF and ROME, we postprocess our images to align with the output of each baseline. For HeadNeRF, we mask both our results and the ground truth results to the non-empty region using the provided HeadNeRF masks, and calculate the Pose and ID losses on the modified images. For ROME, we first downsample both our output and the ground truth images from 512^2 to 256^2 to align with the ROME output, then align the ROME output to the ground truth images using the same landmark detection and Procrustes alignment as described for PTI. Again, we mask both our output and ground truth to the non-empty region predicted by ROME, then calculate Pose and ID on the processed images.

H3DS.. For the depth evaluations on the H3DS dataset [Ramon et al. 2021], we select a frontal image from all 23 subjects, then render the ground truth depth from the corresponding camera pose and the ground truth mesh. We normalize each depth to lie within [0,1]. We then feed the RGB images as input to all baselines, and compute each method's corresponding depth maps. For ours, ours (LT), EG3D-PTI, and ROME, we compute the scale- and translation-invariant L1 and RMSE errors only on the valid depth pixels from the ROME prediction. For HeadNeRF, we use only the valid depth pixels from its prediction. We found that the geometry of HeadNeRF can collapse to a plane in front of the predicted 3D face.

A4 DISCUSSION

Ethical considerations. Since our method does not predict a latent space for portrait editing, it offers limited capabilities for portrait manipulations for malicious uses. However, it may be used to manipulate the viewpoint of a portrait. Potential solutions include detection of unseen image generators [Corvi et al. 2022; Nagano and Luebke 2021] and image watermarking [Yu et al. 2021, 2022].

Adding real images to the training. Intuitively, incorporating real data into the training pipeline may be desirable in order to maximize the photorealism of rendered images and robustness in the most challenging settings. Future work may investigate the best way to use both synthetic and real data in conjunction with one another.

Extension to handling a video input. The framework of our model is such that we require only a single image at inference time. Our single-image method can be extended to handle a video input in a frame by frame fashion, but this may lead to flickering and temporal inconsistency when rendering videos due to the single-image nature of our model. In such cases where multiple images of a subject are available at inference, it is desirable to incorporate all such available information. Further work may investigate making the triplane autoregressive or recurrent, conditioned on the previous frames so that occlusions are handled in a consistent way, and there is greater temporal coherence.

REFERENCES

Adrian Bulat and Georgios Tzimiropoulos. 2017. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In *International Conference on Computer Vision*.

Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. 2018. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 67–74.

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. 2022. Efficient Geometry-aware 3D Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. 2022. On the detection of synthetic images generated by diffusion models.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. 2022. Image Quality Assessment: Unifying Structure and Texture Similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

Pavel Iakubovskii. 2019. Segmentation Models Pytorch. https://github.com/qubvel/segmentation_models.pytorch.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020. Training Generative Adversarial Networks with Limited Data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 2023. 3D GAN Inversion with Pose Optimization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Koki Nagano and David Luebke. 2021. StyleGAN3 Detector. https://github.com/NVlabs/stylegan3-detector.

Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. 2019. Deep face normalization. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–16.

Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. 2021. H3D-Net: Few-Shot High-Fidelity 3D Head Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. *arXiv preprint arXiv:2106.05744* (2021).

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dong Yi, Zhen Lei, Shengcai Liao, and S. Li. 2014. Learning Face Representation from Scratch. *ArXiv* (2014).

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. 2021. Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data. In *IEEE International Conference on Computer Vision (ICCV)*.

Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry S. Davis, and Mario Fritz. 2022. Responsible Disclosure of Generative Models Using Scalable Fingerprinting. In *International Conference on Learning Representations (ICLR)*.