Efficient Geometry-aware 3D Generative Adversarial Networks

Eric R. Chan [*†1,2], Connor Z. Lin[*1], Matthew A. Chan[*1], Koki Nagano[*2], Boxiao Pan[1], Shalini De Mello[2], Orazio Gallo[2], Leonidas Guibas[1], Jonathan Tremblay[2], Sameh Khamis[2], Tero Karras[2], and Gordon Wetzstein[1]

[1]Stanford University [2]NVIDIA

Abstract

Unsupervised generation of high-quality multi-view-consistent images and 3D shapes using only collections of single-view 2D photographs has been a long-standing challenge. Existing 3D GANs are either compute-intensive or make approximations that are not 3D-consistent; the former limits quality and resolution of the generated images and the latter adversely affects multi-view consistency and shape quality. In this work, we improve the computational efficiency and image quality of 3D GANs without overly relying on these approximations. For this purpose, we introduce an expressive hybrid explicit-implicit network architecture that, together with other design choices, synthesizes not only high-resolution multi-view-consistent images in real time but also produces high-quality 3D geometry. By decoupling feature generation and neural rendering, our framework is able to leverage state-of-the-art 2D CNN generators, such as StyleGAN2, and inherit their efficiency and expressiveness. We demonstrate state-of-the-art 3D-aware synthesis with FFHQ and AFHQ Cats, among other experiments.

1. Introduction

Generative adversarial networks (GANs) have seen immense progress, with recent models capable of generating high-resolution, photorealistic images indistinguishable from real photographs [27–29]. Current state-of-the-art GANs, however, operate in 2D only and do not explicitly model the underlying 3D scenes.

Recent work on 3D-aware GANs has begun to tackle the problem of multi-view-consistent image synthesis and, to a lesser extent, extraction of 3D shapes without being supervised on geometry or multi-view image collections. However, the image quality and resolution of existing 3D GANs have lagged far behind those of 2D GANs. Furthermore, their 3D reconstruction quality, so far, leaves much to be desired. One of the primary reasons for this gap is the computational inefficiency of previously employed 3D generators and neural rendering architectures.

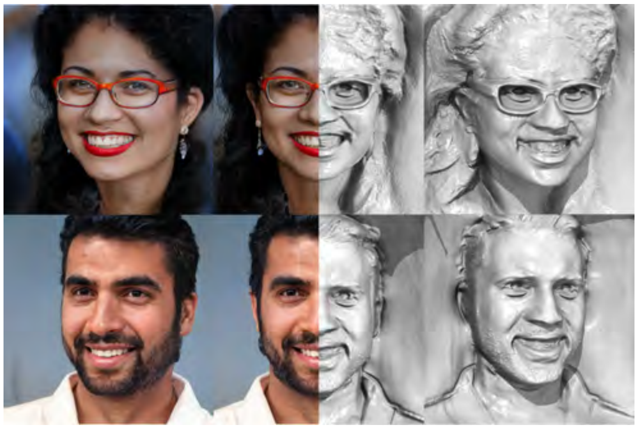


Figure 1. Our 3D GAN enables synthesis of scenes, producing high-quality, multi-view-consistent renderings and detailed geometry. Our approach trains from a collection of 2D images without target-specific shape priors, ground truth 3D scans, or multi-view supervision. Please see the accompanying video for more results.

In contrast to 2D GANs, 3D GANs rely on a combination of a 3D-structure-aware inductive bias in the generator network architecture and a neural rendering engine that aims at providing view-consistent results. The inductive bias can be modeled using explicit voxel grids [14,21,47,48,68,74] or neural implicit representations [4, 47, 49, 58]. While successful in single-scene "overfitting" scenarios, neither of these representations is suitable for training a high-resolution 3D GAN because they are simply too memory inefficient or slow. Training a 3D GAN requires rendering tens of millions of images, but state-of-the-art neural vol-

[*]Equal contribution.

[†]Part of the work was done during an internship at NVIDIA.

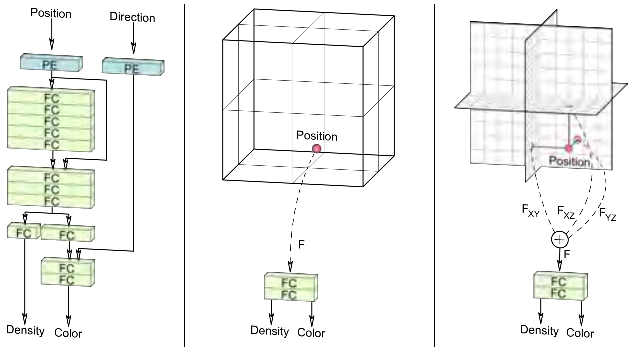
Project page: https://matthew-a-chan.github.io/EG3D ume rendering [45] at high-resolutions with these representations is computationally infeasible. CNN-based image upsampling networks have been proposed to remedy this [49], but such an approach sacrifices view consistency and impairs the quality of the learned 3D geometry.

We introduce a novel generator architecture for unsupervised 3D representation learning from a collection of single-view 2D photographs that seeks to improve the computational efficiency of rendering while remaining true to 3D-grounded neural rendering. We achieve this goal with a two-pronged approach. First, we improve the computational efficiency of 3D-grounded rendering with a hybrid explicit–implicit 3D representation that offers significant speed and memory benefits over fully implicit or explicit approaches without compromising on expressiveness. These advantages enable our method to skirt the computational constraints that have limited the rendering resolutions and quality of previous approaches [4, 58] and forced over-reliance on image-space convolutional upsampling [49]. Second, although we use some image-space approximations that stray from the 3D-grounded rendering, we introduce a dual-discrimination strategy that maintains consistency between the neural rendering and our final output to regularize their undesirable view-inconsistent tendencies. Moreover, we introduce pose-based conditioning to our generator, which decouples pose-correlated attributes (e.g., facial expressions) for a multi-view consistent output during inference while faithfully modeling the joint distributions of pose-correlated attributes inherent in the training data.

As an additional benefit, our framework decouples feature generation from neural rendering, enabling it to directly leverage state-of-the-art 2D CNN-based feature generators, such as StyleGAN2, to generalize over spaces of 3D scenes while also benefiting from 3D multi-view-consistent neural volume rendering. Our approach not only achieves state-of-the-art qualitative and quantitative results for view-consistent 3D-aware image synthesis, but also generates high-quality 3D shapes of the synthesized scenes due to its strong 3D-structure-aware inductive bias (see Fig. 1).

Our contributions are the following:

- We introduce a tri-plane-based 3D GAN framework, which is both efficient and expressive, to enable high-resolution geometry-aware image synthesis.
- We develop a 3D GAN training strategy that promotes multi-view consistency via dual discrimination and generator pose conditioning while faithfully modeling pose-correlated attribute distributions (e.g., expressions) present in real-world datasets.
- We demonstrate state-of-the-art results for unconditional 3D-aware image synthesis on the FFHQ and AFHQ Cats datasets along with high-quality 3D geometry learned entirely from 2D in-the-wild images.



(a) NeRF (Implicit) (b) Voxels (Explicit or Hybrid) (c) Ours (Hybrid)

Figure 2. Neural implicit representations use fully connected layers (FC) with positional encoding (PE) to represent a scene, which can be slow to query (a). Explicit voxel grids or hybrid variants using small implicit decoders are fast to query, but scale poorly with resolution (b). Our hybrid explicit–implicit tri-plane representation (c) is fast and scales efficiently with resolution, enabling greater detail for equal capacity.

We will release source code and pre-trained models.

2. Related work

Neural scene representation and rendering. Emerging neural scene representations use differentiable 3D-aware representations [1, 3, 6, 8, 13, 17, 43, 44, 52, 65] that can be optimized using 2D multi-view images via neural rendering [15, 20, 24, 30, 34–37, 40, 45, 46, 50, 51, 54, 62, 63, 70–72]. Explicit representations, such as discrete voxel grids (Fig. 2b), are fast to evaluate but often incur heavy memory overheads, making them difficult to scale to high resolutions or complex scenes [38, 61]. Implicit representations, or coordinate networks (Fig. 2a), offer potential advantages in memory efficiency and scene complexity compared to discrete voxel grids by representing a scene as a continuous function (e.g., [43, 45, 52, 60, 66]). In practice, these implicit architectures use large fully connected networks that are slow to evaluate as each query requires a full pass through the network. Therefore, fully explicit and implicit representations provide complementary benefits.

Local implicit representations [3, 5, 23, 56] and hybrid explicit–implicit representations [11,35,39,53] combine the benefits of both types of representations by offering computationally and memory-efficient architectures. Inspired by these ideas, we design a new hybrid explicit–implicit 3D-aware network that uses a memory-efficient tri-plane representation to explicitly store features on axis-aligned planes that are aggregated by a lightweight implicit feature decoder for efficient volume rendering (Fig. 2c). Our representation bears some resemblance to previous plane-based hybrid architectures [11, 53], but it is unique in its specific design. Our representation is key to enabling the high 3D GAN image quality that we demonstrate through efficient training comparable (in time scales) to modern 2D GANs [27].

Generative 3D-aware image synthesis. Generative adversarial networks [16] have recently achieved photorealistic image quality for 2D image synthesis [25,28,29,55]. Extending these capabilities to 3D settings has started to gain momentum as well. Mesh-based approaches build on the most popular primitives used in computer graphics, but lack the expressiveness needed for high-fidelity image generation [33, 64]. Voxel-based GANs directly extend the CNN generators used in 2D settings to 3D [14, 21, 47, 48, 68, 74]. The high memory requirements of voxel grids and the computational burden of 3D convolutions, however, make it challenging to adapt them to high-resolution 3D GAN training. Low-resolution 3D volume generation can be remedied with 2D CNN-based image upsampling layers [49], but without an inductive 3D bias the results often lack view consistency. Block-based sparse volume representations partly overcome some of these issues, but are applicable to mostly empty scenes [19, 35] and difficult to generalize across scenes. As an alternative, fully implicit representation networks have been proposed for 3D scene generation [4, 58], but these architectures are slow to query, which makes the GAN training inefficient, limiting the quality and resolution of generated images.

One of the primary insights of our work is that an efficient 3D GAN architecture with 3D-grounded inductive biases is crucial for successfully generating high-resolution view-consistent images and high-quality 3D shapes. Our framework achieves this in several ways. First, unlike most existing 3D GANs, we directly leverage a state-of-the-art 2D CNN-based feature generator, i.e., StyleGAN2 [29], removing the need for inefficient 3D convolutions on explicit voxel grids. Second, our tri-plane representation allows us to leverage neural volume rendering as an inductive bias, but in a computationally much more efficient way than fully implicit 3D networks [4, 45, 58]. Similar to [49], we also employ 2D CNN-based upsampling after neural rendering, but our method introduces dual discrimination to avoid view inconsistencies introduced by the upsampling layers. Unlike existing StyleGAN2-based 2.5D GANs, which generate images and depth maps [59], our method works naturally for steep camera angles and in 360° viewing conditions.

The concurrently developed, but still unpublished 3D-aware GANs StyleNeRF [18] and CIPS-3D [73] demonstrate impressive image quality. The central distinction between these and ours is that while StyleNeRF and CIPS-3D operate primarily in image-space, with less emphasis on the 3D representation, our method operates primarily in 3D. Our approach demonstrates greater view consistency, and is capable of generating high-quality 3D shapes. Furthermore, our experiments report superior FID image scores on FFHQ and AFHQ.

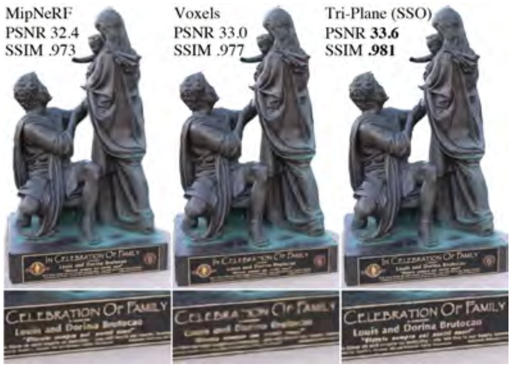


Figure 3. A synthesized view of the multi-view *Family* scene, comparing a fully implicit Mip-NeRF representation (left), a dense voxel grid (center), and our tri-plane representation (right). Even though neither voxels nor tri-planes model view-dependent effects, they achieve high quality.

3. Tri-plane hybrid 3D representation

Training a high-resolution GAN requires a 3D representation that is both efficient and expressive. In this section, we introduce a new hybrid explicit–implicit tri-plane representation that offers both of these advantages. We introduce the representation in this section for a single-scene overfitting (SSO) experiment, before discussing how it is integrated in our GAN framework in the next section.

In the tri-plane formulation, we align our explicit features along three axis-aligned orthogonal feature planes, each with a resolution of $N \times N \times C$ (Fig. 2c) with N being spatial resolution and C the number of channels. We query any 3D position $x \in \mathbb{R}^3$ by projecting it onto each of the three feature planes, retrieving the corresponding feature vector (F_{xy}, F_{xz}, F_{yz}) via bilinear interpolation, and aggregating the three feature vectors via summation. An additional lightweight decoder network, implemented as a small MLP, interprets the aggregated 3D features F as color and density. These quantities are rendered into RGB images using (neural) volume rendering [41, 45].

The primary advantage of this hybrid representation is efficiency—by keeping the decoder small and shifting the bulk of the expressive power into the explicit features, we reduce the computational cost of neural rendering compared to fully implicit MLP architectures [2, 45] without losing expressiveness. To validate that the tri-plane representation is compact yet sufficiently expressive, we evaluate it with a common novel-view synthesis setup. For this purpose, we directly optimize the features of the planes and the weights of the decoder to fit 360° views of a scene from the Tanks & Temples dataset [31] (Fig. 3). In this experiment, we use feature planes of resolution $N = 512$ and channels $C = 48$, paired with an MLP of four layers of 128 hidden units each and a Fourier feature encoding [66]. We compare the results against a dense feature volume of equal capacity. For reference, we include comparisons to a

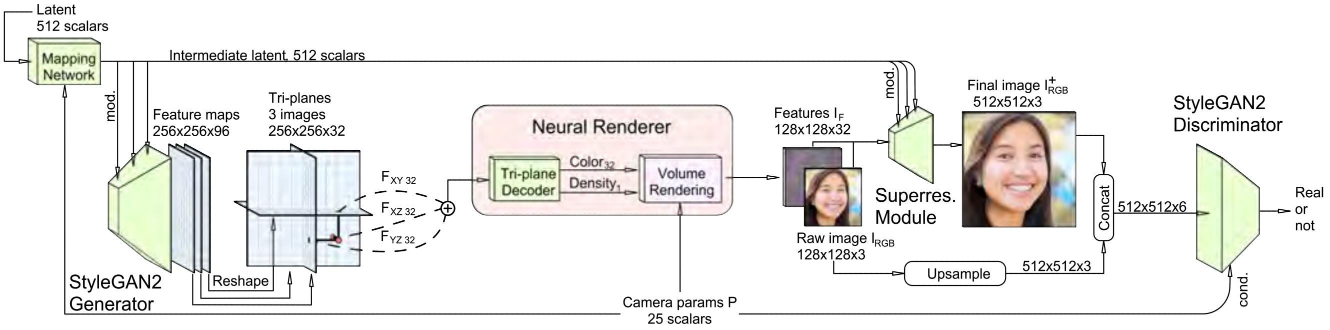


Figure 4. Our 3D GAN framework comprises several parts: a pose-conditioned StyleGAN2-based feature generator and mapping network, a tri-plane 3D representation with a lightweight feature decoder, a neural volume renderer, a super-resolution module, and a pose-conditioned StyleGAN2 discriminator with dual discrimination. This architecture elegantly decouples feature generation and neural rendering, allowing the use of a powerful StyleGAN2 generator for 3D scene generalization. Moreover, the lightweight 3D tri-plane representation is both expressive and efficient in enabling high-quality 3D-aware view synthesis in real-time.

	MLP	Rel. Speed ↑	Rel. Mem. ↓
Mip-NeRF [2]	8×256	1×	1×
Voxels (hybrid)	4×128	3.5×	0.33×
Tri-plane (SSO)	4×128	2.9×	0.32×
Tri-plane (GAN)	1×64	7.8×	0.06×

Table 1. Relative speedups and memory consumption compared to Mip-NeRF. The proposed tri-plane representation is 3–8× faster than a fully implicit Mip-NeRF network and only requires a fraction of its memory. In this example, both voxel grid and tri-plane representation use an MLP-based decoder, as indicated. The number of voxels is chosen to match the total parameters of the tri-plane representation, thus the resolution is relatively low and the memory footprint lower than Mip-NeRF. In the SSO experiment (Fig. 3), we used a larger decoder for the tri-plane representation than for the GAN experiments discussed in Sec. 4 to optimize expressiveness over speed for this experiment.

state-of-the-art fully implicit 3D representation [2]. Fig. 3 and Tab. 1 demonstrate that the tri-plane representation is capable of representing this complex scene, albeit without view-dependent effects, outperforming dense feature volume representations [38, 61] and fully implicit representations [45] in terms of PSNR and SSIM, while offering considerable advantages in computation and memory efficiency. For a side length of N features, tri-planes scale with $O(N^2)$ rather than $O(N^3)$ as dense voxels do, which means for equal capacity and memory, the tri-plane representation can use higher resolution features and capture greater detail. Finally, our tri-plane representation has one other key advantage over these alternatives: the feature planes can be generated with an off-the-shelf 2D CNN-based generator, enabling generalization across 3D representations using the GAN framework discussed next.

4. 3D GAN framework

Armed with an efficient and expressive 3D representation, we tackle training a 3D GAN for geometry-aware image synthesis from a collection of 2D photographs, without any explicit 3D or multi-view supervision. In our setup, each training image is associated with a set of camera intrinsics and extrinsics that we determine at dataset construction time using off-the-shelf pose detectors [10, 32]; see the supplement for details.

Fig. 4 gives an overview of our network architecture. We use the tri-plane representation introduced in the last section to efficiently render images through neural volume rendering, but make a number of modifications to adapt this representation to the 3D GAN setting. Unlike in the SSO experiment, where the features of the planes were directly optimized from the multiple input views, for the GAN setting we generate the tri-plane features, each containing 32 channels, with the help of a 2D convolutional StyleGAN2 backbone (Sec. 4.1). Instead of producing an RGB image, in the GAN setting our neural renderer aggregates features from each of the 32-channel tri-planes and predicts 32-channel feature images from a given camera pose. This is followed by a "super-resolution" module to upsample and refine these raw neurally rendered images (Sec. 4.2). The generated images are critiqued by a slightly modified StyleGAN2 discriminator (Sec. 4.3). The entire pipeline is trained end-to-end from random initialization, using the non-saturating GAN loss function [16] with R1 regularization [42], following the training scheme in StyleGAN2 [29]. The following sections discuss these components in detail. For additional implementation details, hyperparameters, network architecture, and training procedures, please see the supplement.

4.1. CNN generator backbone and rendering

The features of the tri-plane representation, when used in our GAN setting, are generated by a StyleGAN2 CNN generator. The random latent code and camera parameters are first processed by a mapping network to yield an intermediate latent code which then modulates the convolution kernels of a separate synthesis network.

We change the output shape of the StyleGAN2 backbone such that, rather than producing a three-channel RGB image, we produce a $256 \times 256 \times 96$ feature image. This feature image is split channel-wise and reshaped to form three 32-channel planes (see Fig. 4). We choose StyleGAN2 for predicting the tri-plane features because it is a well-understood and efficient architecture achieving state-of-the-art results for 2D image synthesis. Furthermore, our model inherits many of the desirable properties of StyleGAN: a well-behaved latent space that enables style-mixing and latent-space interpolation (see Sec. 5 and supplement).

We sample features from the tri-planes, aggregate by summation, and process the aggregated features with a lightweight decoder, as described in Sec. 3. Our decoder is a multi-layer perceptron with a single hidden layer of 64 units and softmax activation functions. The MLP does *not* use a positional encoding, coordinate inputs, or view-direction inputs. This hybrid representation can be queried for continuous coordinates and outputs a scalar density σ as well as a 32-channel feature, both of which are then processed by a neural volume renderer to project the 3D feature volume into a 2D feature image.

Volume rendering [41] is implemented using two-pass importance sampling as in [45]. Following [49], volume rendering in our GAN framework produces feature images, rather than RGB images, because feature images contain more information that can be effectively utilized for the image-space refinement described next. For the majority of the experiments reported in this manuscript, we render 32-channel feature images I_F at a resolution of 128^2, with 96 total depth samples per ray.

4.2. Super-resolution

Although the tri-plane representation is significantly more computationally efficient than previous approaches, it is still too slow to natively train or render at high resolutions while maintaining interactive framerates. We thus perform volume rendering at a moderate resolution (e.g., 128^2) and rely upon image-space convolutions to upsample the neural rendering to the final image size of 256^2 or 512^2.

Our super-resolution module is composed of two blocks of StyleGAN2-modulated convolutional layers that upsample and refine the 32-channel feature image I_F into the final RGB image I^+_{RGB}. We disable per-pixel noise inputs to reduce texture sticking [27] and reuse the mapping network of the backbone to modulate these layers.

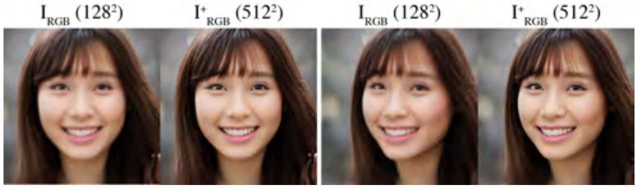


Figure 5. Dual discrimination ensures that the raw neural rendering I_{RGB} and super-resolved output I^+_{RGB} maintain consistency, enabling high-resolution and multi-view-consistent rendering.

4.3. Dual discrimination

As in standard 2D GAN training, the resulting renderings are critiqued by a 2D convolutional discriminator. We use a StyleGAN2 discriminator with two modifications.

First, we introduce *dual discrimination* as a method to avoid multi-view inconsistency issues observed in prior work [47, 49]. For this purpose, we interpret the first three feature channels of a neurally rendered feature image I_F as a low-resolution RGB image I_{RGB}. Intuitively, dual discrimination then ensures consistency between I_{RGB} and the super-resolved image I^+_{RGB}. This is achieved by bilinearly upsampling I_{RGB} to the same resolution as I^+_{RGB} and concatenating the results to form a six-channel image (see Fig. 4). The real images fed into the discriminator are also processed by concatenating each of them with an appropriately blurred copy of itself. We discriminate over these six-channel images instead of the three-channel images traditionally seen in GAN discriminators.

Dual discrimination not only encourages the final output to match the distribution of real images, but also offers additional effects: it encourages the neural rendering to match the distribution of downsampled real images; and it encourages the super-resolved images to be consistent with the neural rendering (see Fig. 5). The second point importantly allows us to leverage effective image-space super-resolution layers without introducing view-inconsistency artifacts.

Second, we make the discriminator aware of the camera poses from which the generated images are rendered. Specifically, following the conditional strategy from StyleGAN2-ADA [26], we pass the rendering camera intrinsics and extrinsics matrices (collectively $\mathbf{P}$) to the discriminator as a conditioning label. We find that this conditioning introduces additional information that guides the generator to learn correct 3D priors. We provide additional studies in the supplement showing the effect of this discriminator conditioning and the robustness of our framework to high levels of noise in the input camera poses.

4.4. Modeling pose-correlated attributes

Most real-world datasets like FFHQ include biases that correlate camera poses and other attributes (e.g., facial expressions), and naively handling them leads to view inconsistent results. For example, the camera angle with respect

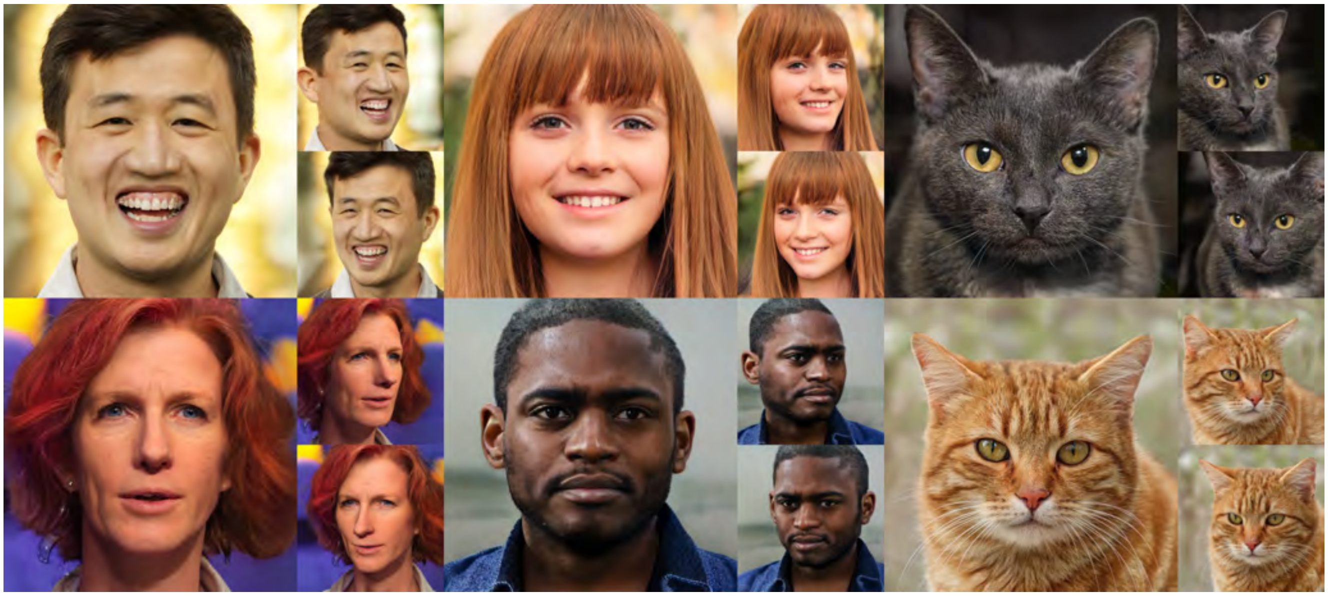


Figure 6. Curated examples at 512^2, synthesized by models trained with FFHQ [28] and AFHQv2 Cats [7]

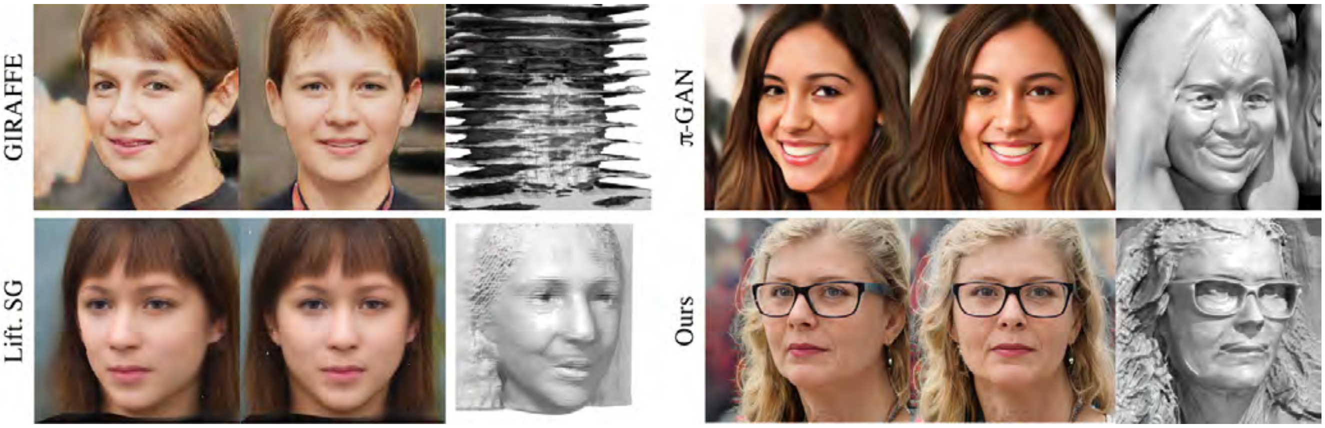


Figure 7. Qualitative comparison between GIRAFFE, pi-GAN, Lifting StyleGAN, ours, with FFHQ at 256^2. Shapes are iso-surfaces extracted from the density field using marching cubes. We inspected the underlying 3D representations of GIRAFFE and found that its over-reliance on image-space approximations significantly harms the learning of the 3D geometry.

to a person's face plays a significant role in whether the person smiles in a photo (see supplement). Similar effects can be observed in other datasets containing only single-view 2D photographs. Faithfully modeling such attribute correlations inherent in the dataset is important for reproducing the best image quality, but at the same time such unwanted attributes need to be decoupled during inference for multi-view consistent synthesis. Related work has been successful at being view consistent [4, 58, 59] or modeling pose-appearance correlations [47, 49], but cannot achieve both simultaneously.

We introduce *generator pose conditioning* as a means to model and decouple correlations between pose and other attributes observed in the training images. To this end, we provide the backbone mapping network not only a latent code vector z, but also the camera parameters $\mathbf{P}$ as input, following the conditional generation strategy in [26]. By giving the backbone knowledge of the rendering camera position, we allow the target view to influence scene synthesis.

During training, pose conditioning allows the generator to model pose-dependent biases implicit to the dataset, allowing our model to faithfully reproduce the image distributions in the dataset. To prevent the scene from shifting with camera pose during inference, we condition the generator on a fixed camera pose when rendering from a moving camera trajectory. We noticed that always conditioning the generator with the rendering camera pose can lead to degenerate solutions where the GAN produces 2D billboards angled towards the camera (please see supplement). To prevent these degenerate solutions, we randomly swap

	FFHQ				Cats
	FID↓	ID↑	Depth↓	Pose↓	FID↓
GIRAFFE 256^2	31.5	0.64	0.94	.089	16.1
π-GAN 128^2	29.9	0.67	0.44	.021	16.0
Lift. SG 256^2	29.8	0.58	0.40	.023	—
Ours 256^2	**4.8**	0.76	**0.31**	**.005**	**3.88**
Ours 512^2	**4.7**	**0.77**	0.39	**.005**	**2.77**†

Table 2. Quantitative evaluation using FID, identity consistency (ID), depth accuracy, and pose accuracy for FFHQ and AFHQ Cats. Labelled is the image resolution of training and evaluation. † Trained with adaptive data augmentation [26].

Res.	GIRAFFE	π-GAN	Lift. SG	Ours	Ours + TC
256^2	181	5	51	27	36
512^2	161	1	—	26	35

Table 3. Runtime in frames per second at different rendering resolutions. We compare variants of our approach with and without tri-plane caching (TC). Run on a single RTX 3090 GPU.

	FID ↓	FACS Smile Std. ↓
Naive model	5.5	0.069
+ DD	6.5	0.054
+ DD, GPC (ours)	**4.7**	**0.031**

Table 4. Dual-discrimination (DD) improves multi-view expression consistency but hurts the model's ability to capture pose-correlated attributes for image quality. Adding generator pose conditioning (GPC) allows the model to improve upon both aspects. Reported at 512^2, with FFHQ.

the conditioning pose in $\mathbf{P}$ with another random pose with 50% probability during training.

5. Experiments and results

Datasets. We compare methods on the task of unconditional 3D-aware generation with FFHQ [28], a real-world human face dataset, and AFHQv2 Cats [7,27], a small, real-world cat face dataset. We augment FFHQ and AFHQv2 with horizontal flips and use off-the-shelf pose estimators [10,32] to extract approximate camera extrinsics with a constant camera focal length that best approximates all images. For all methods on AFHQv2, we apply transfer learning from corresponding FFHQ checkpoints, following [26]; for our method on AFHQv2 512^2, we additionally use adaptive data augmentation [26]. For more results, please see the accompanying video.

5.1. Comparisons

Baselines. We compare our methods against three state-of-the-art methods for 3D-aware image synthesis: π-GAN [4], GIRAFFE [49], and Lifting StyleGAN [59].

Qualitative results. Fig. 6 presents selected examples synthesized by our model with FFHQ and AFHQ at a resolution of 512^2, highlighting the image quality, view-consistency, and diversity of outputs produced by our method. Fig. 7 provides a qualitative comparison against baselines. While GIRAFFE synthesizes high-quality images, reliance on view-inconsistent convolutions produces poor-quality shapes and identity shift—note the hairline inconsistency between rendered views. π-GAN and Lifting StyleGAN generate adequate shapes and images but both struggle with photorealism and in capturing detailed shapes.

Our method synthesizes not only images that are higher quality and more view-consistent but also higher-fidelity 3D geometry as seen in the detailed glasses and hair strands.

Quantitative evaluations. Table 2 provides quantitative metrics comparing the proposed approach against baselines. We measure image quality with Fréchet Inception Distance (FID) [22] between 50k generated images and all available real images. We evaluate shape quality by calculating MSE against pseudo-ground-truth depth-maps (Depth) and poses (Pose) estimated from synthesized images by [10]; a similar evaluation was introduced by [59]. We assess multi-view facial identity consistency (ID) by calculating the mean Arcface [9] cosine similarity score between pairs of views of the same synthesized face rendered from random camera poses. Additional evaluation details are provided in the supplement. Our model demonstrates significant improvements in FID across both datasets, bringing the 3D GAN to near the same level as StyleGAN2 512^2 (2.97 for FFHQ [29] and 2.99 for Cats [26]) while also maintaining state-of-the-art view consistency, geometry quality, and pose accuracy.

Runtime. Table 3 compares rendering speed at inference running on a single NVIDIA RTX 3090 GPU. Our end-to-end approach achieves real-time framerates at 512^2 final resolution with 128^2 neural rendering resolution and 96 total depth samples per ray, suitable for applications such as real-time visualization. When rendering consecutive frames of a static scene, we need not regenerate the tri-plane features every frame; caching the generated features is a simple tweak that improves render speed. The proposed approach is significantly faster than fully implicit alternatives, such as π-GAN [4]. Although it is more computationally expensive than Lifting StyleGAN [59] and GIRAFFE [49], we believe major improvements in image quality, geometry quality, and view-consistency outweigh the increased compute cost.

5.2. Ablation study

Without dual discrimination, generated images can include multi-view inconsistencies due to the unconstrained image-space super-resolution layers. We measure this effect quantitatively by extracting a few smile-related Facial Action Coding System (FACS) [12] coefficients from videos produced by models with and without dual discrimination,

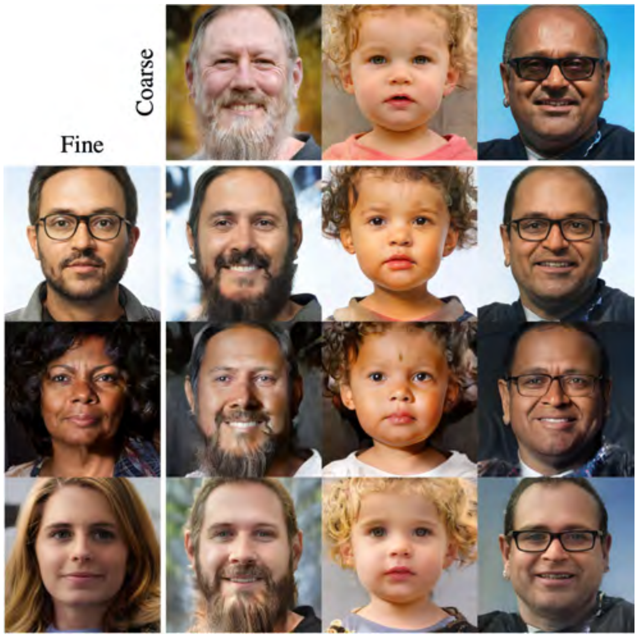


Figure 8. Style-mixing [27–29] examples from a model trained on FFHQ 512^2, without truncation.

using a proprietary facial tracker. We measure the standard deviation of smile coefficients for the same scene across video frames. A view-consistent scene should exhibit little expression shift and thus produce little variation in smile coefficients. This is validated in Table 4 showing that introducing dual discrimination (second row) reduces the smile coefficient variation versus the naive model (first row), indicating improved expression consistency. However, dual discrimination also reduces image quality as seen by the slightly worse FID score, perhaps because the model is restricted from reproducing the pose-correlated attribute bias observed in the FFHQ dataset. By adding generator pose conditioning (third row), we allow the generator to faithfully model pose-correlated attributes in the dataset while decoupling them at inference, leading to both the best FID score and view-consistent results.

5.3. Applications

Style mixing. Since our 3D representation is designed with the StyleGAN2 backbone from the ground up, it inherits the well-studied properties of the StyleGAN2 latent space, allowing us to do semantic image manipulations. Fig. 8 shows our method's results for style mixing [27–29].

Single-view 3D reconstruction. Fig. 9 shows the application of our learned latent space for single-view 3D reconstruction. We use pivotal tuning inversion (PTI) [57] to fit test images. The learned 3D prior over FFHQ enables surprisingly high-quality single-view geometry recovery, even for an out-of-domain grayscale input image. Further exploration of few-shot 3D reconstruction and novel-view-synthesis may prove a fruitful avenue for future work.

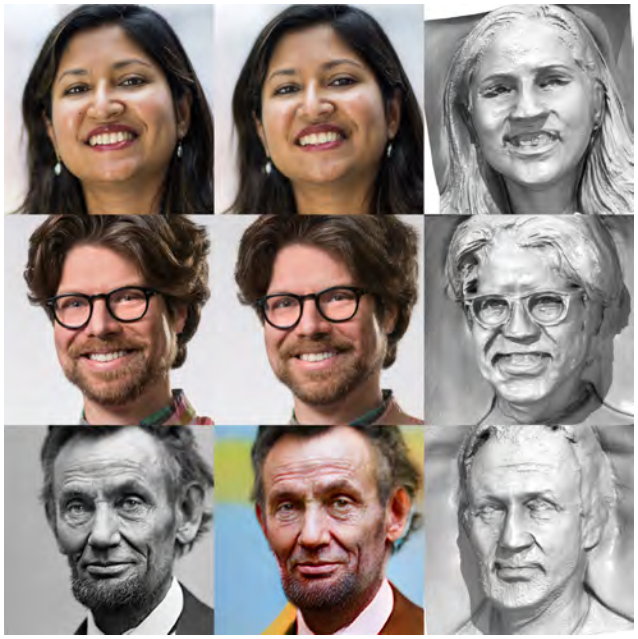


Figure 9. We use PTI [57] to fit a target image and recover the underlying 3D shape. Target (left); reconstructed image (center); reconstructed shape (right). From a model trained on FFHQ 512^2.

6. Discussion

Limitations and future work. Although our shapes show significant improvements over those generated by previous 3D-aware GANs, they still lack finer details, such as individual teeth, and certain artifacts in extracted shapes remain. To further improve the quality of the learned shapes, we could instill a stronger geometry prior or regularize the density component of the radiance field following methods proposed by [51, 67, 69].

Our model requires knowledge of the camera pose distribution of the dataset. Although prior work has proposed learning the pose distribution on the fly [49], others have noticed such methods can diverge [18], so it would be fruitful to explore this direction further. Pose conditioning aids the generator in decoupling appearance with pose, but it still does not fully disentangle the two. Furthermore, ambiguities that can be explained by geometry remain unresolved. For example, by creating concave eye sockets, the generator creates the illusion of eyes that "follow" the camera, an incorrect interpretation, though the renderings are view-consistent and reflect the underlying geometry.

In principle, it is possible to use any 2D backbone using our framework. As such, we believe that the exploration of alternative backbones, such as image-to-image translation or Transformer-based approaches, could enable new applications in conditional synthesis.

Ethical considerations. The single-view 3D reconstruction or style mixing applications could be misused for generating edited imagery of real people. Such misuse of image synthesis techniques poses a societal threat, and we do not condone using our work with the intent of spreading misinformation or tarnishing reputation. We also recognize a potential lack of diversity in our faces results, stemming from implicit biases of the datasets we process.

Conclusion. By combining an efficient explicit–implicit neural representation with an expressive pose-aware convolutional generator and a dual discriminator, our approach takes significant steps towards photorealistic 3D-aware image synthesis and high-quality unsupervised shape generation. This may enable rapid prototyping of 3D models, more controllable image synthesis, and novel techniques for shape reconstruction from temporal data.

Acknowledgements

We thank David Luebke, Jan Kautz, Jaewoo Seo, Jonathan Granskog, Simon Yuen, Alex Evans, Stan Birchfield, Alexander Bergman, and Joy Hsu for reviewing early drafts and for helpful suggestions and feedback. We thank Alex Chan, Giap Nguyen, and Trevor Chan for help with figures and diagrams. 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.

References

[1] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In ICCV, 2021. 3, 4

[3] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In European Conference on Computer Vision (ECCV), 2020. 2

[4] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 1, 2, 3, 6, 7

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 6, 7

[8] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. Overfit neural networks as a compact shape representation. arXiv preprint arXiv:2009.09808, 2020. 2

[9] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In CVPR, 2019. 7

[10] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In IEEE Computer Vision and Pattern Recognition Workshops, 2019. 4, 7

[11] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. arXiv preprint arXiv:2104.00670, 2021. 2

[12] Paul Ekman and Wallace V. Friesen. Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, 1978. 7

[13] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. Science, 2018. 2

[14] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In International Conference on 3D Vision, 2017. 1, 3

[15] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200fps. arXiv preprint arXiv:2103.10380, 2021. 2

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems (NeurIPS), 2014. 3, 4

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In International Conference on Machine Learning (ICML), 2020. 2

[18] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. arXiv preprint arXiv:2110.08985, 2021. 3, 8

[19] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. GANcraft: Unsupervised 3D neural rendering of minecraft worlds. In IEEE International Conference on Computer Vision (ICCV), 2021. 3

[20] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In IEEE International Conference on Computer Vision (ICCV), 2021. 2

[21] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping Plato's cave: 3D shape from adversarial rendering. In IEEE International Conference on Computer Vision (ICCV), 2019. 1, 3

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. In Advances in Neural Information Processing Systems (NeurIPS), 2017. 7

[23] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3D scenes. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[24] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In International Conference on Learning Representations (ICLR), 2018. 3

[26] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 5, 6, 7

[27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In Advances in Neural Information Processing Systems (NeurIPS), 2021. 1, 2, 5, 7, 8

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 1, 3, 6, 7, 8

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1, 3, 4, 7, 8

[30] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[31] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics, 36(4), 2017. 3

[32] Taehee Brad Lee. Cat hipsterizer, 2018. https://github.com/kairess/cat_hipsterizer. 4, 7

[33] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3D controllable image synthesis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 3

[34] David B Lindell, Julien NP Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[35] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 2, 3

[36] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3D supervision. arXiv preprint arXiv:1911.00767, 2019. 2

[37] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: Rendering deep implicit signed distance function with differentiable sphere tracing. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[38] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. ACM Transactions on Graphics (SIGGRAPH), 2019. 2, 4

[39] Julien N.P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. ACORN: Adaptive coordinate networks for neural representation. ACM Transactions on Graphics (SIGGRAPH), 2021. 2

[40] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[41] N. Max. Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics (TVCG), 1995. 3, 5

[42] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In International Conference on Machine Learning (ICML), 2018. 4

[43] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[44] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In IEEE International Conference on Computer Vision (ICCV), 2019. 2

[45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In European Conference on Computer Vision (ECCV), 2020. 2, 3, 4, 5

[46] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. Computer Graphics Forum, 40(4), 2021. 2

[47] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In IEEE International Conference on Computer Vision (ICCV), 2019. 1, 3, 5, 6

[48] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. BlockGAN: Learning 3D object-aware scene representations from unlabelled images. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 1, 3

[49] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 1, 2, 3, 5, 6, 7, 8

[50] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[51] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In IEEE International Conference on Computer Vision (ICCV), 2021. 2, 8

[52] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[53] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In European Conference on Computer Vision (ECCV), 2020. 2

[54] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. arXiv preprint arXiv:2011.13961, 2020. 2

[55] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In International Conference on Learning Representations (ICLR), 2016. 3

[56] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. In IEEE International Conference on Computer Vision (ICCV), 2021. 2

[57] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. arXiv preprint arXiv:2106.05744, 2021. 8

[58] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 1, 2, 3, 6

[59] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2D stylegan for 3D-aware face generation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 3, 6, 7

[60] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 2

[61] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. DeepVoxels: Learning persistent 3D feature embeddings. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2, 4

[62] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In Advances in Neural Information Processing Systems (NeurIPS), 2019. 2

[63] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[64] Attila Szabó, Givi Meishvili, and Paolo Favaro. Unsupervised generative 3D shape learning from natural images. arXiv preprint arXiv:1910.00287, 2019. 3

[65] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[66] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 2, 3

[67] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. Advances in Neural Information Processing Systems (NeurIPS), 2021. 8

[68] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In Advances in Neural Information Processing Systems (NeurIPS), 2016. 1, 3

[69] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. arXiv preprint arXiv:2106.12052, 2021. 8

[70] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 2

[71] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In IEEE International Conference on Computer Vision (ICCV), 2021. 2

[72] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492, 2020. 2

[73] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. arXiv preprint arXiv:2110.09788, 2021. 3

[74] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, and William T.

Freeman. Visual object networks: Image generation with disentangled 3D representations. In Advances in Neural Information Processing Systems (NeurIPS), 2018. 1, 3

Supplemental Material
Efficient Geometry-aware 3D Generative Adversarial Networks

Eric R. Chan *†1,2, Connor Z. Lin*1, Matthew A. Chan*1, Koki Nagano*2, Boxiao Pan1, Shalini De Mello2, Orazio Gallo2, Leonidas Guibas1, Jonathan Tremblay2, Sameh Khamis2, Tero Karras2, and Gordon Wetzstein1

1Stanford University 2NVIDIA

In this supplement, we first provide additional experiments (Section 1) and visual results (Section 2). We follow with details of our implementation (Section 3), including further descriptions of model architecture and training process, as well as hyperparameters. We discuss experiment details (Section 4), such as datasets and baselines, and further explanations for experiments such as inversion. Lastly, we consider artifacts (Section 5) that may be targets of future work. We encourage readers to view the accompanying supplemental video, which contains additional visual results, including a live demonstration of real-time synthesis.

1. Additional experiments

1.1. Analyzing pose/facial expression correlation in FFHQ

Figure 1 plots the likelihood a subject from FFHQ [16] is smiling (measured by [35]), against head yaw (computed by [9]). The plot indicates that individuals facing towards the camera are more likely to be smiling than are individuals who are facing away from the camera. An intuitive explanation for this phenomenon is that people who are knowingly being photographed, as in portrait images, are more likely to be smiling than people who are photographed candidly.

Left uncompensated for, this correlation between pose and facial expressions incentivizes "expression warping", where the expressions of synthesized faces shift as we move the camera. We propose dual discrimination (Section 4.3 of the main paper) and generator pose conditioning (Section 4.4 of the main paper) to reduce such expression warping.

1.2. COLMAP reconstruction

To further validate the multi-view consistency of our method, we employ COLMAP [29, 30] to reconstruct a point-cloud of a synthesized video sequence (Figure 2). We reconstruct a video sequence of 128 frames, taken from an oval trajectory similar to the camera paths shown in the supplemental video. We use COLMAP's "automatic" reconstruction, without specifying camera parameters. The re-

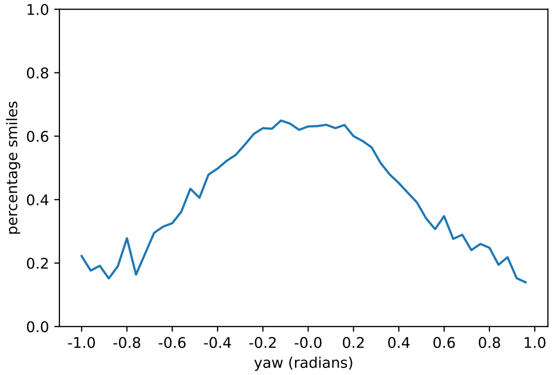


Figure 1. We plot the probability of smiling against head yaw angle, as measured by [35]. People looking at the camera are more likely to be smiling than people angled away, indicating a correlation between scene appearance and camera pose.

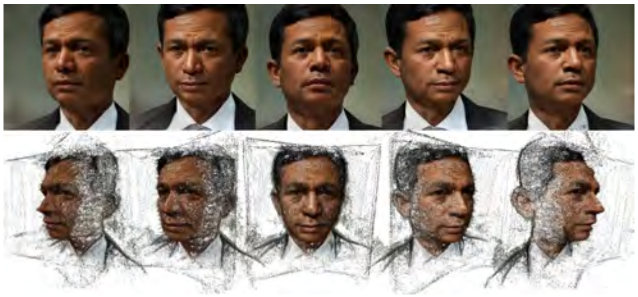


Figure 2. COLMAP [29, 30] reconstruction of 128 frames of synthesized video (top) which followed an oval trajectory. The resulting dense, well-defined point cloud (bottom) is indicative of highly multi-view-consistent rendering.

[*] Equal contribution.

[†] Part of the work was done during an internship at NVIDIA.

sulting point cloud is dense and well-defined, indicating that our 3D GAN produces highly multi-view-consistent renderings.

1.3. Regularizing generator pose conditioning

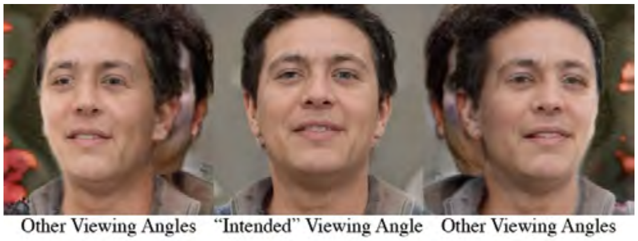


Figure 3. Naively applying generator pose conditioning results in a degenerate solution because the generator is always aware of the location of the rendering camera. Such an approach produces reasonable renderings when taken from the "intended" viewing angle, (i.e. the camera pose the generator was conditioned on). However, if we freeze the conditioning information and move the camera at inference, it is clear that the model has learned to produce "billboards" angled towards the known location of the camera.

As described in Section 4.4 of the main paper, we regularize generator pose conditioning by randomly swapping the conditioning pose of the generator with another random pose with 50% probability. Figure 3 shows the result of training a model with generator pose conditioning but without any swapping regularization—the generator always receives, as a conditioning input, the true pose of the rendering camera. The model learns a degenerate solution in which it creates a "billboard" angled towards the rendering camera. We prevent this degenerate solution by randomly swapping the conditioning camera pose with an alternative pose sampled from the dataset pose distribution. For models shown, we swap the conditioning vector with 100% probability at the start of training; the swapping probability is linearly decayed to 50% over the first 1M images. For the remainder of training, we maintain 50% swapping probability.

1.4. Robustness to imprecise camera poses

Our method expects a dataset in which each image is labeled with an approximate camera pose, in order to enable sampling camera poses from the dataset distribution and discriminator pose conditioning. While such labelling can be easily performed with pre-trained pose extractors on humans [9] and cats [19], extracting accurate poses may be difficult for some datasets. This section evaluates reliance on discriminator pose conditioning and on accurate camera poses. We train five additional models on FFHQ 256^2: a "baseline" configuration without discriminator pose conditioning, and four discriminator-pose-conditioned models where camera poses are corrupted with increasing levels of random noise. We calculate the 4×4 standard deviation matrix, σ, by taking the standard deviation across the dataset of ground-truth 4×4 camera pose matrices. We train four models with "imprecise" camera poses: ($1\,\sigma$, $2\,\sigma$, $3\,\sigma$, $4\,\sigma$) where the input camera poses matrices are corrupted with 1, 2, 3, and 4 standard deviations of Gaussian noise, respectively. We train these five ablations on FFHQ 256^2 with a shortened training curriculum of 4M images, in order to save computational resources.

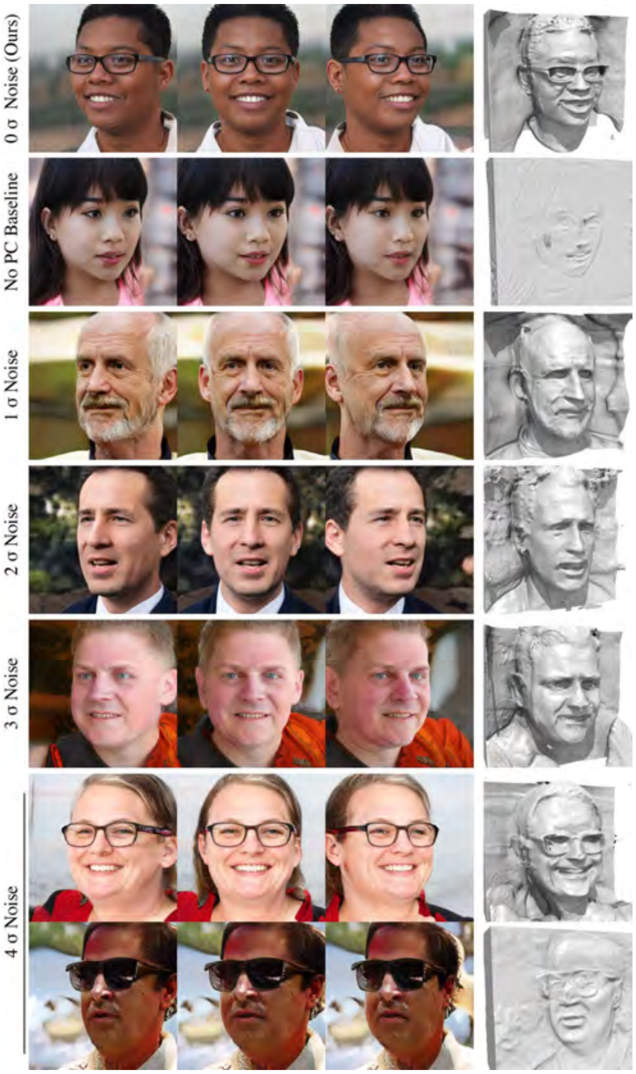


Figure 4. In order to gauge robustness to the accuracy of the supplied camera poses, we compare a baseline without discriminator pose conditioning against discriminator-pose-conditioned models where camera extrinsics are corrupted by noise. Without discriminator pose conditioning, the model learns a degenerate solution in which heads are drawn as a texture flattened onto a plane. Even highly imprecise extrinsics (e.g. camera poses corrupted by three standard deviations of noise) are capable of resolving this degenerate solution and allow recovery of accurate 3D shapes.

Figure 4 shows the results of this experiment. Without discriminator pose conditioning, the model falls into a

	FFHQ					Cats		Cars	
	FID↓	KID↓	ID↑	Depth↓	Pose↓	FID↓	KID↓	FID↓	KID↓
GIRAFFE 128^2	—	—	—	—	—	—	—	27.3	1.703
GIRAFFE 256^2	31.5	1.992	0.64	0.94	.089	16.1	2.723	—	—
π-GAN 128^2	29.9	3.573	0.67	0.44	.021	16.0	1.492	17.3	0.932
Lift. SG 256^2	29.8	—	0.58	0.40	.023	—	—	—	—
Ours 128^2	—	—	—	—	—	—	—	**2.75**	**0.097**
Ours 256^2	**4.8**	**0.149**	0.76	**0.31**	**.005**	**3.88**	**0.091**	—	—
Ours 512^2	**4.7**	**0.132**	**0.77**	0.39	**.005**	**2.77**†	**0.041**†	—	—

Table 1. Quantitative evaluation using FID, KID×100, identity consistency (ID), depth accuracy, and pose accuracy for FFHQ [16] and FID, KID×100 for AFHQv2 Cats [7,15] and ShapeNet Cars [6,32]. Labeled is the image resolution of training and evaluation. † Trained with adaptive discriminator augmentation [14].

degenerate solution in which it renders textures on a flat plane, without properly capturing the 3D shape of scenes. Providing even very imprecise camera poses is enough to break this tendency; conditioning the discriminator on camera poses distorted by three standard deviations of Gaussian noise still produces accurate 3D shapes. With extreme noise (e.g. four standard deviations), some scenes maintain the correct 3D structure while others are flattened onto the plane. Our results indicate that while our method requires additional information to prevent collapse, only very weak supervision is necessary. Future work may examine this tendency further and discover ways to prevent this undesirable behavior without requiring images to be labelled with poses.

1.5. Extrapolation to steep camera angles

Figure 5 provides a visual comparison of our method against baselines for generating views from steep camera poses. We note that the FFHQ [16] dataset is primarily composed of front-facing images—few images depict faces from extreme yaw angles, and even fewer images depict faces from extreme pitch angles. Nevertheless, reasonable extrapolation to the edges of the pose distribution is a desirable quality and indicates reliance on a robust 3D representation.

Lifting StyleGAN [31], which represents scenes as a textured mesh, demonstrates consistent rendering quality. However the steep camera angles reveal inaccurate 3D geometry (e.g. foreshortened faces) learned by the method. π-GAN [5], reasonably extrapolates to steep angles but exhibits visible quality degradation at the edges of the pose distribution. GIRAFFE [26], being highly reliant on view-inconsistent convolutions, has difficulty reproducing angles that are rarely seen in the dataset. If we force GIRAFFE to extrapolate beyond the camera poses sampled at training (e.g. the leftmost and rightmost images of Fig. 5b), we receive degraded, view-inconsistent images rather than renderings from steeper angles. The problem is amplified for pitch (Fig. 5a) because the dataset's pitch range is even narrower.

Our method, despite also using 2D convolutions, is less reliant on view-inconsistent convolutions for considering the placement of features in the final image. By utilizing an expressive 3D representation as a "scaffold", our method provides more reasonable extrapolation to rare views in both pitch and yaw than methods that more strongly depend on image-space convolutions for image synthesis, such as GIRAFFE [26].

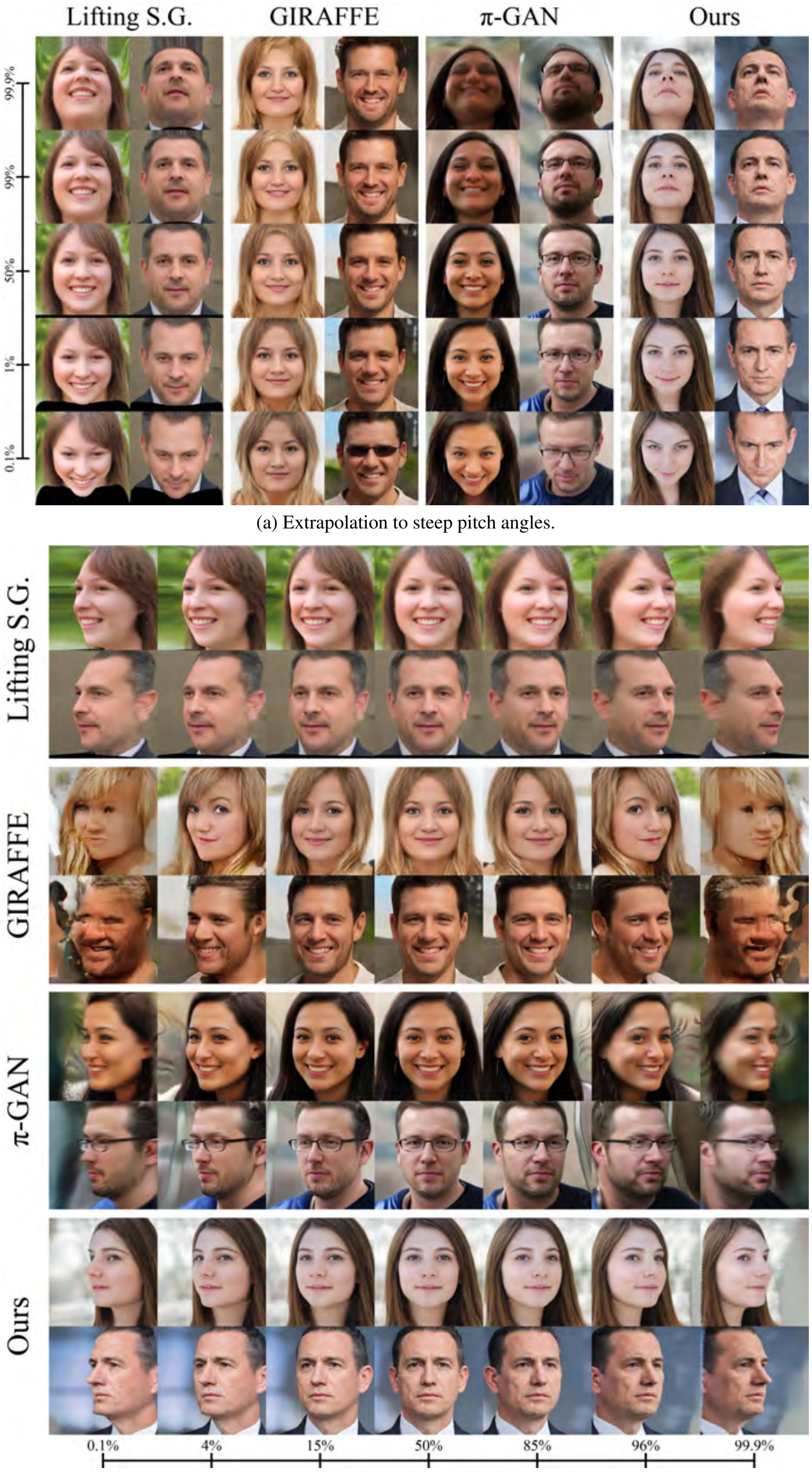
1.6. Additional quantitative results

Table 1 is an expanded version of Table 2 of the main manuscript that provides additional quantitative metrics, including Kernel Inception Distance [2] for all datasets and image quality evaluations for ShapeNet Cars. Strong relative performance on Cars, a dataset in which camera poses are distributed uniformly about the sphere, is evidence that our method is not restricted to face-forward datasets like FFHQ [16] and AFHQv2 [7, 15].

2. Additional visual results

Style mixing, in shapes. Figure 6 shows the underlying shapes of the style mixing [16] examples in Figure 8 of the main manuscript. While mixed examples inherit most of their shape structure from the modulations of the backbone's low-resolution layers, the modulations of the high-resolution layers can influence fine details in the shape, such as eye regions and hair patterns. The results were obtained from a model trained without style-mixing regularization.

Additional single image 3D reconstructions. Figure 7 provides additional 3D reconstructions of single test images through Pivotal Tuning Inversion (PTI) [28] of a model trained on FFHQ 512^2. A pipeline for high-fidelity, single-image reconstruction of faces that does not require explicit 3D ground-truth training data opens the door for many



(a) Extrapolation to steep pitch angles.

(b) Extrapolation to steep yaw angles.

Figure 5. We compare methods in their extrapolation to steep camera viewing angles. Labelled is the percentile for camera pitch or yaw. A yaw angle in the 96th percentile means 96% of training poses are less steep, i.e. 4% of training poses are beyond the given pose.

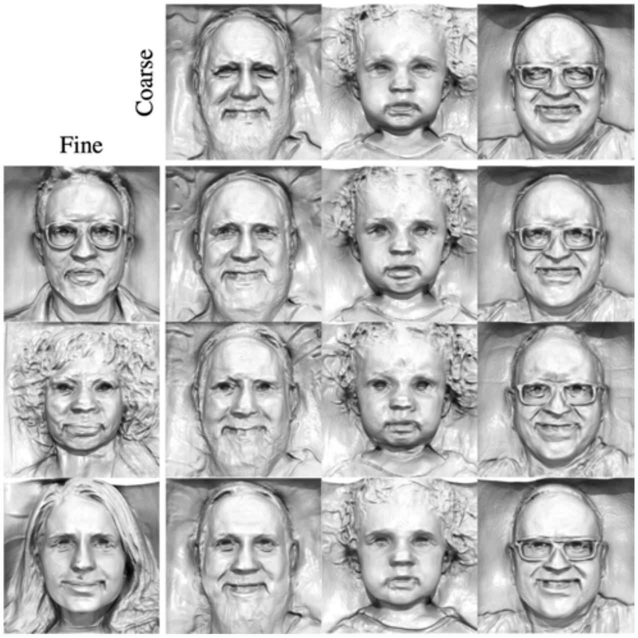


Figure 6. Style-mixing [15–17] shapes from a model trained on FFHQ 512^2, without truncation. Aligns with Figure 8 of the main manuscript, which shows color renderings of the same seeds. The result illustrates that while a mixed example inherits the majority of its structure from its "coarse" input (i.e. modulations of layers 0-6), the "fine" input (i.e. modulations of layers 7-13) can influence the more delicate details of the shape (e.g. eye regions, hair patterns), in addition to having much control over the overall colors in rendered images.

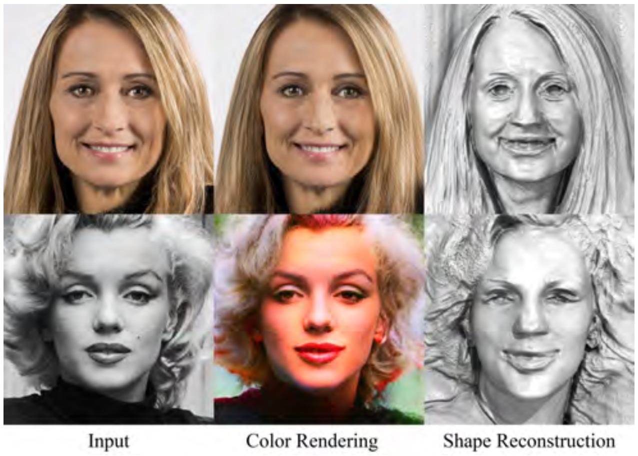


Figure 7. Additional single-view 3D reconstructions of test images demonstrate a use for our generator's learned prior over facial features.

promising applications, such as photo-to-avatar creation.

Shapenet Cars. Figure 8 contains uncurated renderings from random camera poses for models trained with ShapeNet Cars [6, 32]. This experiment serves as a demonstration that our method is capable of operating successfully on datasets that include camera poses that span the entire 360° camera azimuth and 180° camera elevation distributions, unlike 2.5D GANs [31], which are intended for face-forward datasets.

Additional selected examples synthesized with AFHQv2 Cats. Figure 9 shows renderings and shapes for selected examples, synthesized by our method trained on AFHQv2 Cats [7, 15] 512^2.

Uncurated examples synthesized with AFHQv2 Cats. Figure 10 provides uncurated examples of cats produced by GIRAFFE [26], π-GAN [5], and our method, trained at image rsolutions of 256^2, 128^2, and 512^2, respectively.

Uncurated examples synthesized with FFHQ. Figure 11 provides uncurated examples of faces produced by our method, trained with FFHQ [16] 512^2. We apply truncation [4, 16, 22], with $\psi = 0.5$.

Latent code interpolation. Figure 12 provides linear interpolations between latent codes for selected examples produced by our method trained on FFHQ 512^2. Our result illustrates that our 3D GAN inherits the well-behaved latent space of the StyleGAN2 [17] backbone, which enables smooth interpolations in both color renderings and underlying shapes.

Additional selected examples synthesized with FFHQ Figure 13 depicts renderings and shapes for selected examples, synthesized by our method trained on FFHQ 512^2.

3. Implementation details

We implemented our 3D GAN framework on top of the official PyTorch implementation of StyleGAN2, an updated version of which is available at https://github.com/NVlabs/stylegan3. Most of our training parameters are identical to those of StyleGAN2 [17], including the use of equalized learning rates for the trainable parameters [13], a minibatch standard deviation layer at the end of the discriminator [13], exponential moving average of the generator weights, and a non-saturating logistic loss [11] with R1 regularization [23].

Two-stage training. In order to save computational resources, we perform the majority of the training at a neural rendering resolution of 64^2, before gradually stepping the resolution up to 128^2. Note that the final image resolution remains fixed throughout training (e.g. 256^2 or 512^2). We implement this simply by bilinearly resizing the raw neural rendering I_{RGB} to 128^2 before it is operated on by the

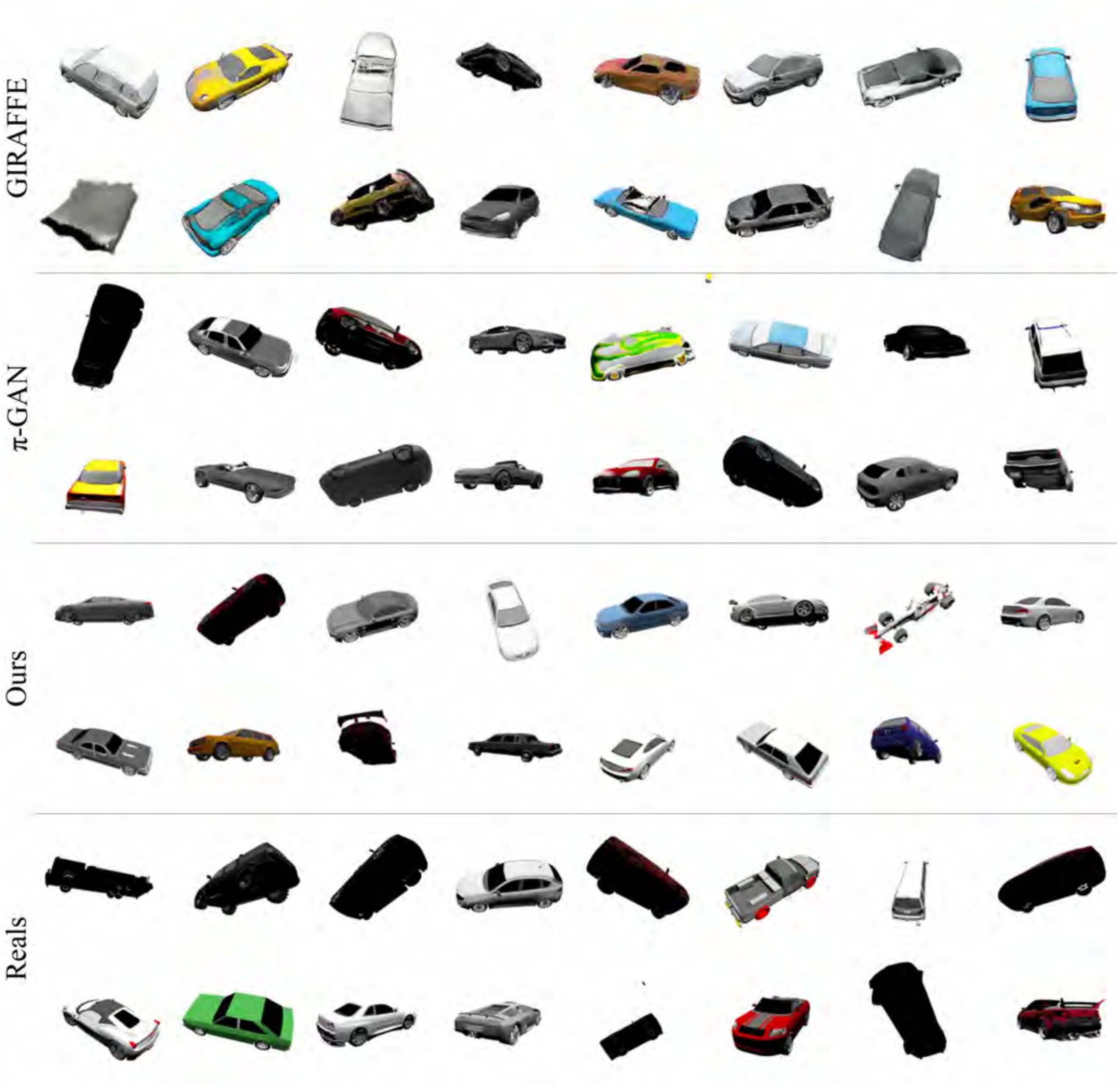


Figure 8. Qualitative comparison of uncurated examples of cars. All methods are sampled with truncation [4, 16, 22], using $\psi = 0.7$.

super-resolution module. Thus, the super-resolution module always receives a 128^2-sized feature map as an input, regardless of the actual neural rendering resolution. In contrast to previous progressive growing strategies [5, 13] that double the resolution in a single step, we gradually increase the neural rendering resolution, pixel-by-pixel, over 1 million images, i.e., (64^2, 65^2, 66^2, ..., 126^2, 127^2, 128^2). We continue training with the resolution fixed at 128^2 for an additional 1.5 million images, for a total of 2.5M iterations of fine-tuning. This two-stage training procedure provides a roughly 2× speed-up versus training from scratch at full resolution and produces similar results to training at full neural rendering resolution from scratch.

Backbone. Our backbone (i.e., StyleGAN2 generator) follows the implementation of [17], with a mapping network of 8 hidden layers. For all of our experiments (regardless of final image resolution), the backbone operates at a resolution of 256^2. We modify the output convolutions such that they produce a 96-channel output feature image, which

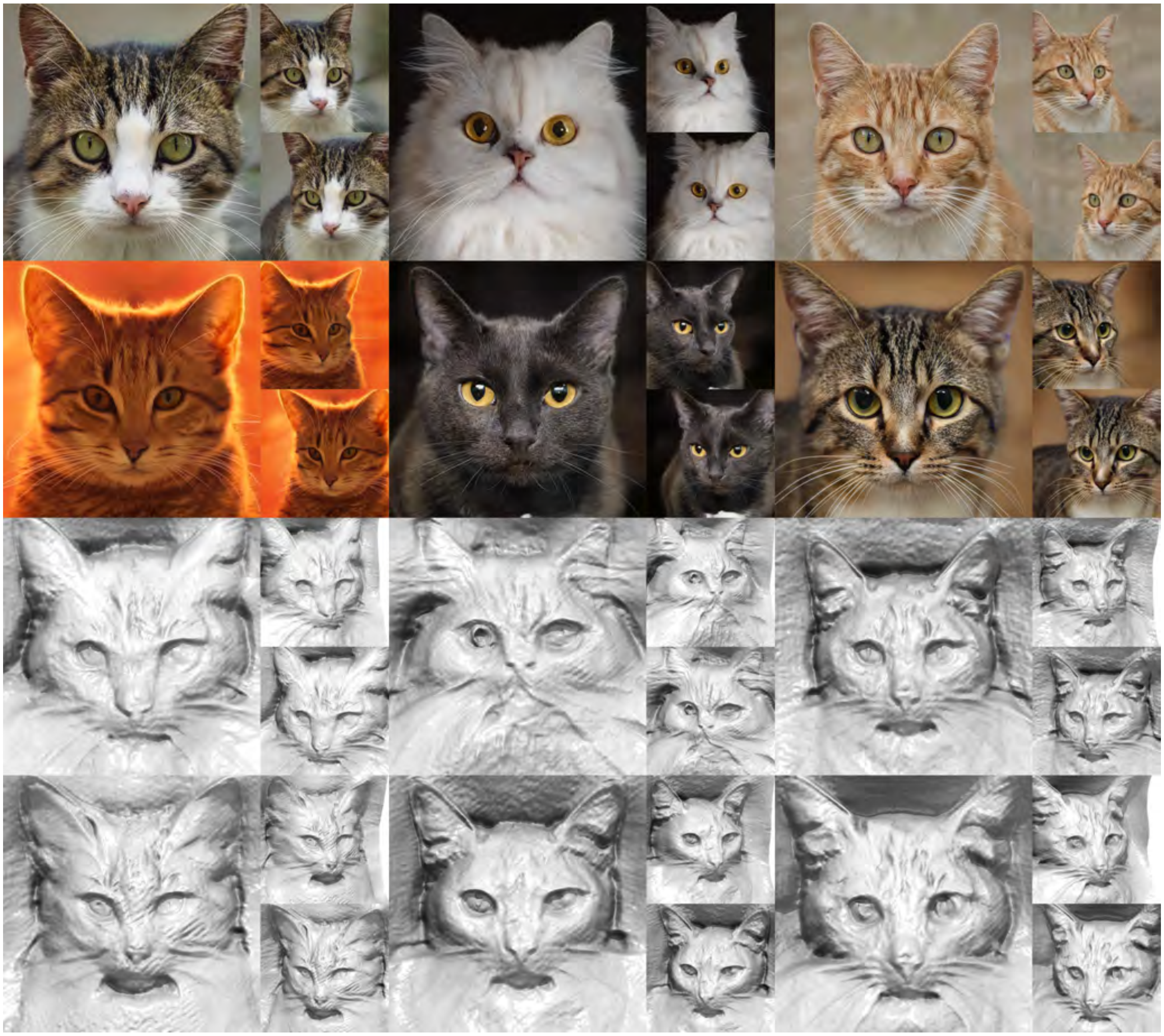


Figure 9. Curated examples from a model trained on AFHQv2 [7, 15] 512^2.

we reshape into three planes, each of shape $256 \times 256 \times 32$. Unlike approaches that require pre-trained 2D image GANs [31], we do not utilize pre-trained StyleGAN2 checkpoints for the backbone; the entire pipeline is trained end-to-end. For large datasets, such as FFHQ [16] and ShapeNet Cars [6, 32], we train from scratch with random initialization; for small datasets, such as AFHQv2 [7, 15], we follow prevailing methodology [14] by fine-tuning from a checkpoint trained on a larger dataset.

Decoder and volume rendering. Our decoder is implemented as an MLP with a single hidden layer of 64 hidden units and uses the softmax activation function. The decoder takes as input a 32-channel aggregated feature vector; it produces a 33-channel vector that we split into a scalar density prediction and a 32-channel feature. We use neural volume rendering [24] of features [26], with two-pass importance sampling. For FFHQ [16] and AFHQv2 [7, 15], we use 48 uniformly-spaced and 48 importance samples per ray; for ShapeNet Cars, we use 64 uniformly-spaced and 64 importance samples per ray. When rendering videos that feature thin surfaces, we found it beneficial to increase the samples per ray during inference to reduce flicker.

Super-resolution. We implement our super-resolution model with two 'blocks' of StyleGAN2's modulated convo-

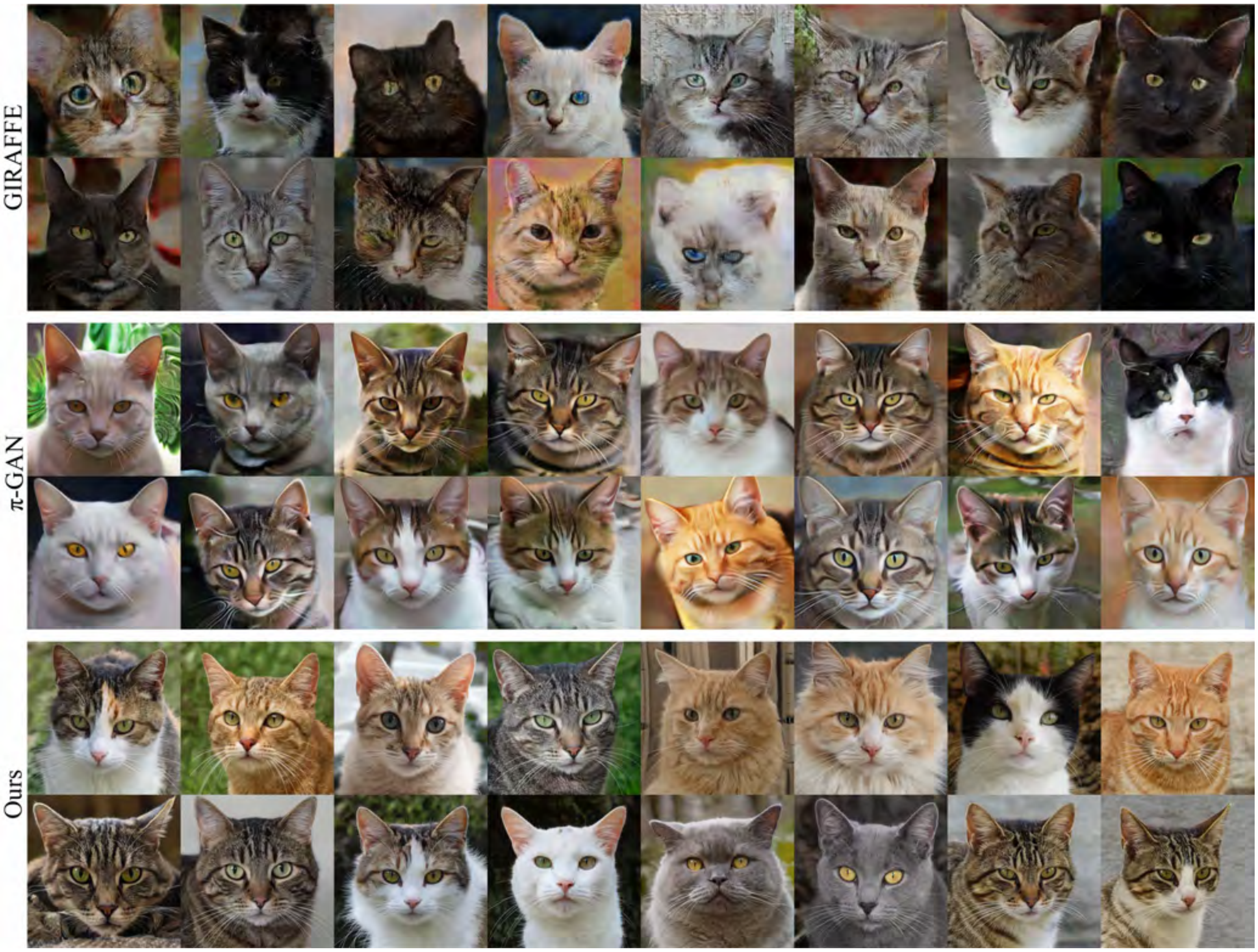


Figure 10. Uncurated examples of cats, for GIRAFFE [26] 256^2, π-GAN 128^2, and our method 512^2. All methods are sampled with truncation [4, 16, 22], using $\psi = 0.7$.

lutions [17], with noise inputs disabled. The blocks contain convolutions of channel-depth 128 and 64, respectively.

Discriminator. Our discriminator is a StyleGAN2 [17] with two modifications. First, to enable dual discrimination, we adjust the input layer to accept six-channel input images, rather than 3-channel input images. Figure 14 provides a diagram that illustrates the creation of these six-channel inputs, for both real and generated images. Second, we condition the discriminator on the camera parameters of the incoming image to help prevent degenerate shape solutions; we follow the class-conditional discriminator modifications of [14] to inject this information.

Mixed Precision. To speed up training, we use a similar mixed-precision methodology as [14]. We use FP16 in the four highest resolution blocks of the discriminator and in both blocks of our super-resolution module. We do not use FP16 in our generator backbone.

R1 Regularization. We use R1 regularization [23] with $\gamma = 1$ for all datasets and resolutions, except for ShapeNet Cars, where we use $\gamma = 0.1$. Regularization strengths were informally chosen based on values that have shown success with previous methods [14, 17].

Training. We train all models with a batch size of 32. We use a discriminator learning rate of 0.002 and a generator learning rate of 0.0025. Following [15], we blur images as they enter the discriminator, gradually reducing the blur amount over the first 200K images. Unlike [17], we train without style-mixing regularization.

Using the two-stage training discussed previously, we train at a resolution of 64^2 for 25M images and at 128^2 for

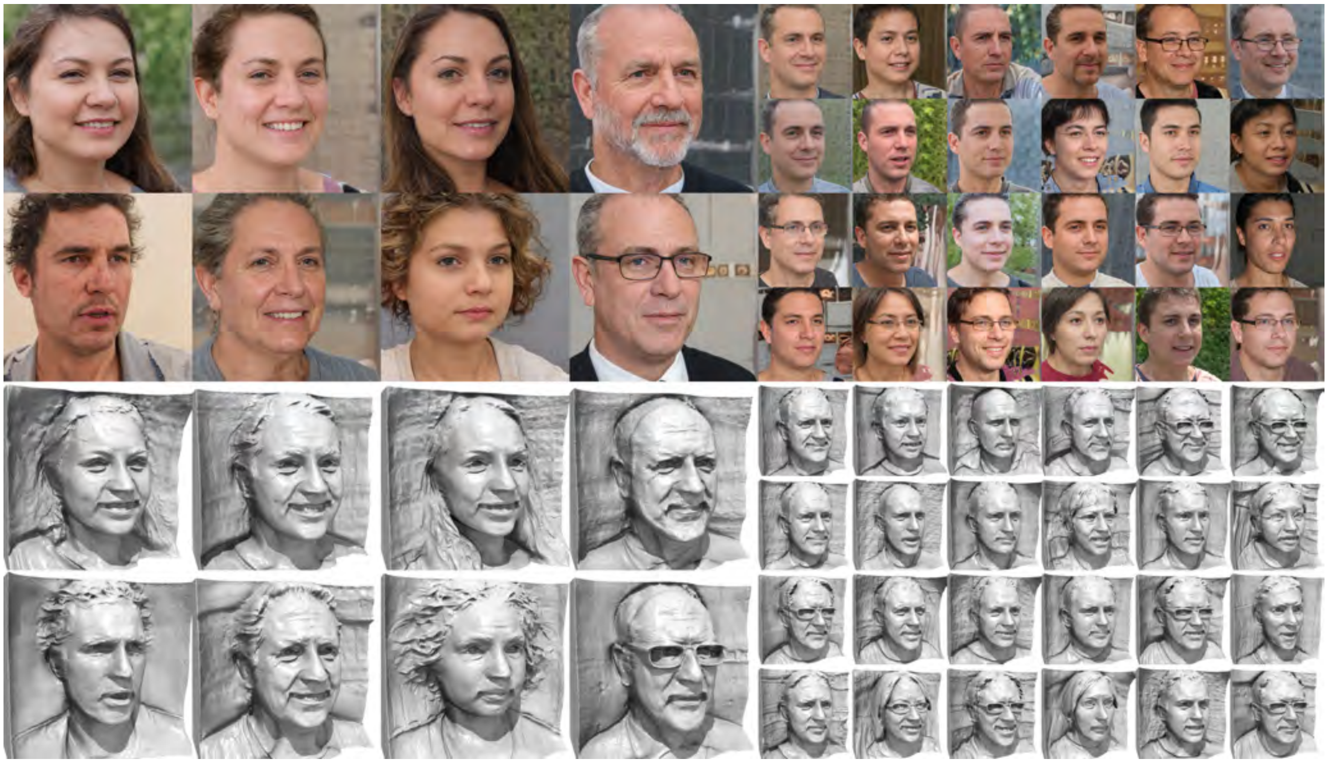


Figure 11. Images and geometry for seeds 0-31, synthesized using a model trained on FFHQ [16] 512^2. Sampled with truncation [16], using $\psi = 0.5$.

an additional 2.5M images. Using a neural rendering resolution of 64^2, our 3D GAN framework takes ∼24 seconds to train on 1000 images (24 s/kimg) on 8 Tesla V100 GPUs; this increases to 46 s/kimg at a neural rendering resolution of 128^2. For reference, StyleGAN3-R [15] achieves training rates of 20 s/kimg on similar hardware.

Our total training time on 8 Tesla V100 GPUs is on the order of 8.5 days (7 days of 64^2 training, plus 1.5 days of 128^2 fine-tuning), compared to 6 days on similar hardware for StyleGAN3-R.

Inference-time depth samples. We use neural volume rendering [24] with two-pass importance sampling to render feature images from our tri-plane representation. We found that increasing the number of samples per ray at inference time can reduce unwanted flickering when rendering videos that feature thin objects such as eye glasses. For clips shown in the supplemental video, we double both the number of coarse samples (from 48 to 96) and the number of fine samples (from 48 to 96), bringing the total number of depth samples per ray to 192. Increasing the number of samples per ray incurs a penalty to the rendering speed. When using 96 total depth samples per ray, frame rates are reduced to approximately 24 frames per second with tri-plane caching – down from 36 frames per second when using the default 48 samples. Images shown in the main manuscript were synthesized without increasing the number of depth samples along each ray.

AFHQv2. Following [14], we fine-tune from FFHQ-trained models to achieve optimum performance on Cats. Beginning from a checkpoint trained on FFHQ, we train for 6.2M images at a neural rendering resolution of 64^2; and for an additional 2.6M images, while fine-tuning the neural rendering resolution to 128^2. Because π-GAN and GIRAFFE were not designed with the benefits of adaptive discriminator augmentation (ADA) [14], we also do not use ADA for our method at 256^2, in an effort to keep comparisons across methods fair. We use adaptive discriminator augmentation with its default settings, for our method only at 512^2 .

4. Experiment details

4.1. Baselines

π-GAN [5] is a 3D-aware GAN that relies upon a FiLM-conditioned MLP with periodic activation functions for camera-controllable synthesis. We utilized the official code (https://github.com/marcoamonteiro/pi-GAN) and trained until convergence with the parameters recommended for analogous datasets.

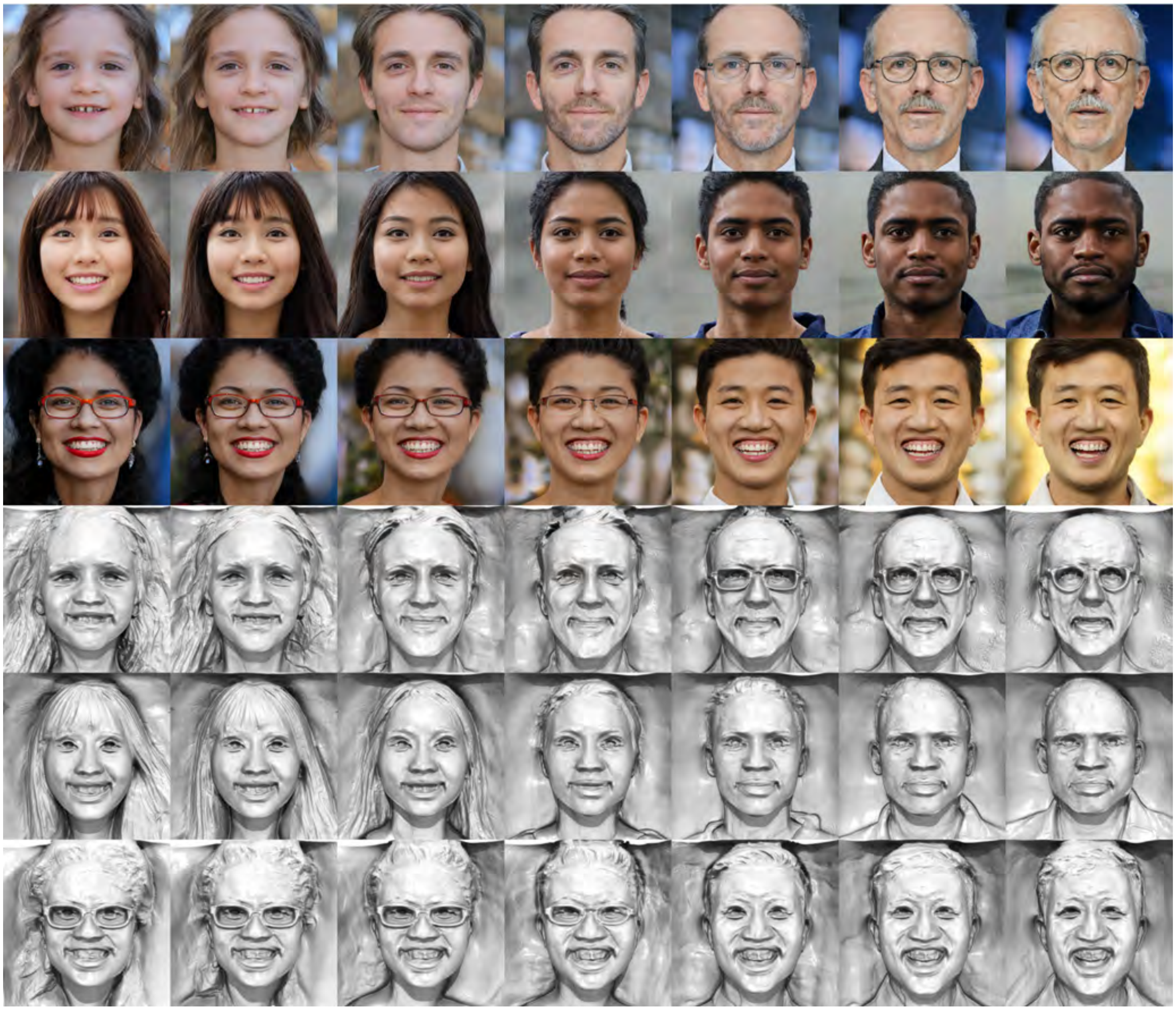


Figure 12. Linear interpolations between latent codes, showing renderings and shapes.

GIRAFFE [26] is a 3D-aware GAN that incorporates a compositional 3D scene representation to enable controllable synthesis. We utilized the official code (https://github.com/autonomousvision/giraffe) and trained until convergence with the parameters recommended for analogous datasets.

Lifting StyleGAN [31] is a method for disentangling and lifting a pre-trained StyleGAN2 image generator to 3D-aware face generation. The original Lifting StyleGAN manuscript reports results on a slightly tighter crop of FFHQ than we used. Because we had difficulty matching the quality of Lifting StyleGAN's pre-trained model when we trained it from scratch on our less-cropped dataset, we instead used their official pre-trained model for their tighter crops and the FID score reported in their manuscript. We utilized the offical code, found here: (https://github.com/seasonSH/LiftedGAN).

StyleGAN2 is a style-based GAN that achieves state-of-the-art image quality for 2D image synthesis and features a well-behaved latent space that enables image manipulation. We obtained a pre-trained checkpoint for StyleGAN2 on FFHQ 512^2 from the collection of official models (https://catalog.ngc.nvidia.com/orgs/nvidia/teams/research/models/stylegan2). Following the recommended tuning of [14], we trained both StyleGAN2 config F and the 512×512 config from [14], sweeping R1 [23] regularization strength, $\gamma = \{0.2, 0.5, 1, 2, 5, 10, 20\}$. The best result for AFHQv2 was

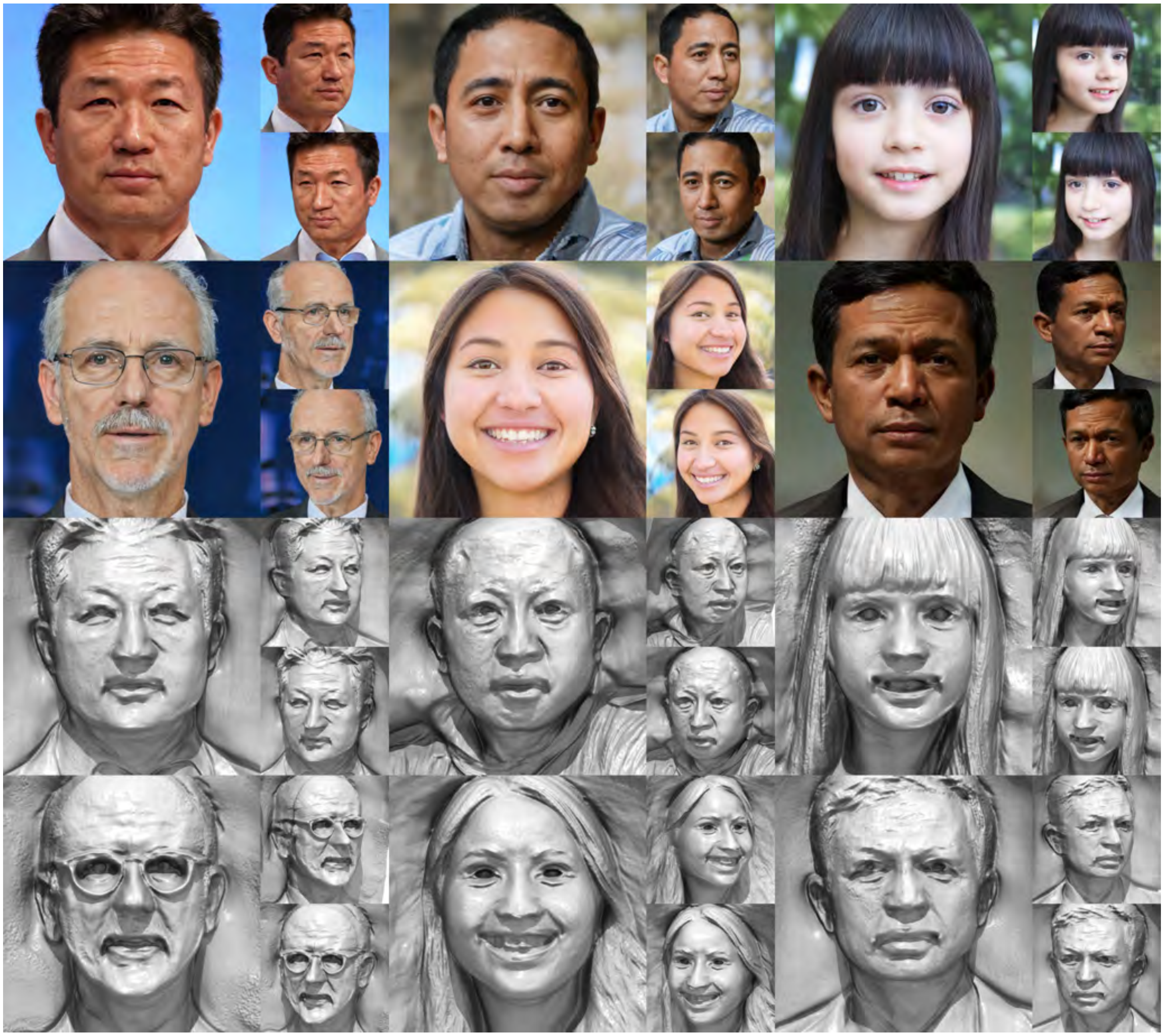


Figure 13. Additional selected examples, from a model trained on FFHQ [16] 512^2.

obtained with StyleGAN2 config F, after training for 10M images at $\gamma = 1$.

4.2. Dataset Details

FFHQ We prepare our dataset by starting with the unaligned FFHQ dataset (FFHQ-U) [15], which is composed of uncropped, unaligned images of people. We use an off-the-shelf face detection and pose-extraction pipeline [9] to both identify the face region and label the image with a pose. We crop the images to roughly the same size as the original FFHQ dataset.

We assume fixed camera intrinsics across the entire dataset, with a focal length of $4.26 \times image\_width$, equivalent to a standard portrait lens. We prune a small number of images that resisted face detection; our final dataset contains 69957 images. We augment the dataset with horizontal flips.

AFHQv2 We used the AFHQv2 dataset [15], which is a higher-quality version of the original AFHQ dataset [7]. AFHQv2 provides closeups for animal faces including cats, dogs, and wildlife. We use the 'cats' split, which contains approximately 5000 images, for our experiments. As with FFHQ, we assume fixed camera intrinsics across the dataset; for simplicity, we use identical intrinsics to FFHQ. Camera poses were extracted via landmark detection [19]

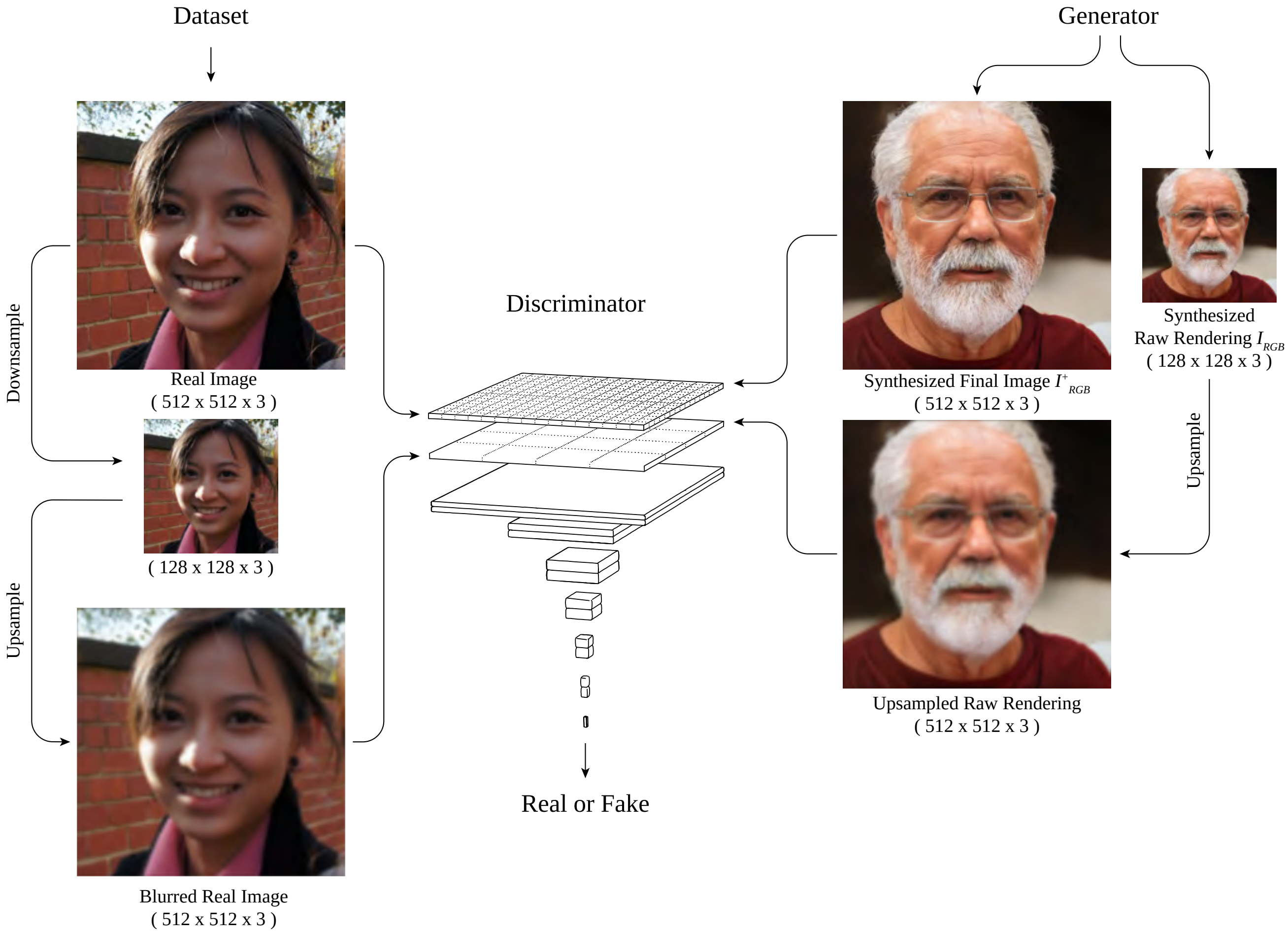


Figure 14. In dual-discrimination, we discriminate on a six-channel concatenation of the final image and the raw neural rendering, in order to maintain consistency between high-resolution final images and view-consistent (but low resolution) neural renderings. This diagram illustrates how we obtain a six-channel discriminator input tensor for both real and fake images. Our generator produces both a 512^2 final rendering (I^+_{RGB}) as well as the (128^2) raw neural rendering (I_{RGB}). The raw rendering, I_{RGB} is the first three channels of the 32-channel rendered features, I_F. We create a six-channel discriminator input by upsampling the raw image to 512^2 and concatenating it with the final image to form a ($512 \times 512 \times 6$) discriminator input tensor. For real images, we extract a 512^2 real image from the dataset and downsample it to the same size as I_{RGB} to obtain an analogue for I_{RGB}. We then upsample this image back to 512^2 and concatenate it with the original image to form a ($512 \times 512 \times 6$) discriminator input tensor. The downsample-then-upsample operation has the effect of blurring the original image.

and an open-source Perspective-n-Point algorithm [3]. We augment the dataset with horizontal flips.

ShapeNet Cars For additional validation, we compare methods on ShapeNet Cars [6, 32] to evaluate performance on a dataset that contains views from all angles. We adopted the dataset and setup from [32], which is composed of 128^2 resolution renderings of synthetic cars, each labelled with camera parameters. The dataset contains 2457 unique cars; each car is rendered from 50 views randomly sampled from the entire sphere. We use the known camera parameters for each image and do not augment the dataset with image space augmentations.

4.3. Single scene overfitting.

To illustrate the effectiveness of our architecture, we evaluate the relative performance of the tri-plane 3D representation against a comparable voxel-based hybrid representation and Mip-NeRF [1] on the *Family* scene of Tanks & Temples [18] dataset as desribed in Section 3 of the main manuscript. We use the pre-processed images, as well as the training/test split, of [21]. We use 512 uniformly-spaced depth samples and 256 importance samples per ray and a ray batch size of 6400. The tri-planes are treated as learnable parameters of shape $3\times48\times512\times512$. The dense voxel parameters were chosen to optimize quality for comparable parameter count as the tri-planes; the voxel features are of shape $18 \times 128 \times 128 \times 128$. Both voxel and tri-plane hybrid representations are coupled with two-layer, 128 hidden unit decoders with Fourier feature embeddings [33]. We train voxel and cube representations for 200K iterations; we train Mip-NeRF for the recommended 1M iterations.

4.4. Pivotal tuning inversion.

We use off-the-shelf face detection [9] to extract appropriately-sized crops and camera extrinsics from test images and we resize each cropped image to 512^2. We follow Pivotal Tuning Inversion (PTI) [28], optimizing the latent code for 500 iterations, followed by fine-tuning the generator weights for an additional 500 iterations.

For inversion of grayscale images, we convert the generator's 3-channel, RGB renderings to perceived luminance, Y, before computing image distance loss during optimization. This allows the generator's prior to colorize the renderings. To compute single-channel luminance from 3-channel RGB images, we use $Y = 0.299R + 0.587G + 0.114B$. For grayscale optimization, we use 400 latent code inversion steps and 250 generator fine-tuning steps.

4.5. Evaluation Metrics

FID and KID. We compute Fréchet Inception Distance (FID) [12] and Kernel Inception Distance (KID) [2] image quality metrics between 50k generated images and all training images using the implementation provided in the StyleGAN3 [15] codebase.

Geometry. We follow a similar procedure to [31] in the evaluation of geometry. We generate 1024 images and depth maps from random poses that match the dataset pose distribution. With the application of a pre-trained 3D face reconstruction model [9], we generate a "pseudo" ground-truth depth map for each generated image. Next we limit both the generated depth maps and "pseudo" ground-truth depth maps to the facial regions as defined by the reconstruction model. Finally, we normalize all depth maps to zero mean, unit variance and calculate the L2 distance between them.

Multi-view consistency. We evaluate multi-view consistency and face identity preservation for models trained on FFHQ [16] by measuring ArcFace [8] cosine similarity. For each method, we generate 1024 random faces and render two views of each face from poses randomly selected from the training dataset pose distribution. For each image pair, we measure facial identity similarity [8] and compute the mean score.

Pose accuracy. We evaluate pose accuracy with the help of a pre-trained face reconstruction model [9]. With [9], we detect pitch, yaw, and roll from 1024 generated images then compute L2 loss against the ground truth poses to determine each model's pose drift.

Runtime. We evaluate runtime for each model by calculating the average framerate over a 400 frame sequence. We process frames consecutively, i.e., with batch size 1. In order to give each method a best-case-scenario, we ignore operations such as copying rendered frames from GPU to CPU and saving files to disk.

FACS estimation In Section 5.2 of the main paper, we quantitatively measure the effect of dual discrimination and generator pose conditioning at preserving facial expressions across multi-view face videos. To evaluate facial expressions, we employ a proprietary facial tracker that measures detailed movement of sub-regions of the face in terms of Facial Action Coding System (FACS) [10] coefficients. Specifically, our facial tracker measures all 53 FACS blendshape coefficients defined in Li et al. [20] and we compared the variability in the 'mouthSmile_L' and 'mouthSmile_R' blendshape coefficients across the different videos.

5. Discussion

5.1. Shape artifacts

Despite significant improvements in the quality of the 3D geometry compared to previous methods, our synthesized shapes are not free from artifacts, which are visible in geometry renderings throughout the main paper and supplement (e.g. Fig. 11, Fig. 13). Sunken eye sockets allow the illusion of eyes that follow the viewing camera, even when the geometry and neural renderings are view-consistent; such "hollow face illusions" have demonstrated similar effects in the physical world. Similarly, deep creases near the corners of mouths enable the creation of "view-inconsistent" effects that in fact are faithful to the underlying shapes. Future work that incorporates stronger dataset priors, e.g. that eyeballs are convex, may help resolve these artifacts.

While our method produces more-detailed eyeglasses than previous methods, it tends to produce "goggles"—the sides of the eyeglasses are opaque where there should be empty space. Future neural rendering methods that can accurately model lens refraction may enable more faithful reconstruction of eyeglasses and other objects that contain transparent elements.

In some shapes and renderings generated by our method, a seam is visible between the face and the rest of the head. We hypothesize that recent hybrid-SDF rendering solutions [27, 34, 36], which have shown promising results in robust geometry recovery from images, may yield improved shapes with fewer artifacts.

In the interests of simplicity, we model the scene with a single 3D representation, without any explicit background handling. Consequently, the generator learns to represent backgrounds of images with textured surfaces fused to foreground objects. Future work that models backgrounds with a separate 3D representation [25, 26, 37] may enable isolation of foreground objects.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In ICCV, 2021. 12

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In International Conference on Learning Representations (ICLR), 2018. 3, 13

[3] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000. 12

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In International Conference on Learning Representations (ICLR), 2019. 5, 6, 8

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 3, 5, 6, 9

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015. 3, 5, 7, 12

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 3, 5, 7, 11

[8] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In CVPR, 2019. 13

[9] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In IEEE Computer Vision and Pattern Recognition Workshops, 2019. 1, 2, 11, 13

[10] Paul Ekman and Wallace V. Friesen. Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, 1978. 13

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems (NeurIPS), 2014. 5

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a nash equilibrium. In Advances in Neural Information Processing Systems (NeurIPS), 2017. 13

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In International Conference on Learning Representations (ICLR), 2018. 5, 6

[14] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 3, 7, 8, 9, 10

[15] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In Advances in Neural Information Processing Systems (NeurIPS), 2021. 3, 5, 7, 8, 9, 11, 13

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 1, 3, 5, 6, 7, 8, 9, 11, 13

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 5, 6, 8

[18] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics, 36(4), 2017. 12

[19] Taehee Brad Lee. Cat hipsterizer, 2018. https://github.com/kairess/cat_hipsterizer. 2, 11

[20] Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, et al. Learning formation of physically-based face attributes. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 13

[21] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 12

[22] Marco Marchesi. Megapixel size image creation using generative adversarial networks, 2017. 5, 6, 8

[23] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In International Conference on Machine Learning (ICML), 2018. 5, 8, 10

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In European Conference on Computer Vision (ECCV), 2020. 7, 9

[25] Michael Niemeyer and Andreas Geiger. CAMPARI: Camera-aware decomposed generative neural radiance fields. arXiv preprint arXiv:2103.17269, 2021. 14

[26] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 3, 5, 7, 8, 10, 14

[27] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In IEEE International Conference on Computer Vision (ICCV), 2021. 13

[28] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. arXiv preprint arXiv:2106.05744, 2021. 3, 13

[29] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 1

[30] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In European Conference on Computer Vision (ECCV), 2016. 1

[31] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2D stylegan for 3D-aware face generation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 3, 5, 7, 10, 13

[32] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In Advances in Neural Information Processing Systems (NeurIPS), 2019. 3, 5, 7, 12

[33] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In Advances in Neural Information Processing Systems (NeurIPS), 2020. 13

[34] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. Advances in Neural Information Processing Systems (NeurIPS), 2021. 13

[35] Jie Wu. Facial expression recognition pytorch, 2018. https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch. 1

[36] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. arXiv preprint arXiv:2106.12052, 2021. 13

[37] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. arXiv:2010.07492, 2020. 14