FaceLit: Neural 3D Relightable Faces

Anurag Ranjan[⌂]   Kwang Moo Yi[⌂][⊡]   Jen-Hao Rick Chang[⌂]   Oncel Tuzel[⌂]

[⌂]Apple   [⊡]The University of British Columbia

Figure 1. Generated samples from our model. **Left**: 3D reconstruction visualization. **Center and Right**: Rendered faces using 2 different illumination conditions under 3 different poses. Illumination visualization using spherical harmonics [26].

Abstract

We propose a generative framework, FaceLit, capable of generating a 3D face that can be rendered at various user-defined lighting conditions and views, learned purely from 2D images in-the-wild without any manual annotation. Unlike existing works that require careful capture setup or human labor, we rely on off-the-shelf pose and illumination estimators. With these estimates, we incorporate the Phong reflectance model in the neural volume rendering framework. Our model learns to generate shape and material properties of a face such that, when rendered according to the natural statistics of pose and illumination, produces photorealistic face images with multiview 3D and illumination consistency. Our method enables photorealistic generation of faces with explicit illumination and view controls on multiple datasets – FFHQ, MetFaces and CelebA-HQ. We show state-of-the-art photorealism among 3D aware GANs on FFHQ dataset achieving an FID score of 3.5.

1. Introduction

Learning a 3D generative model from 2D images has recently drawn much interest [13, 23, 32]. Since the introduction of Neural Radiance Fields (NeRF) [21], the quality of images rendered from a 3D model [6, 13] has improved drastically, becoming as photorealistic as those rendered by a 2D model [19]. While some of them [7, 28] rely purely on 3D representations to deliver 3D consistency and pay the price of decreased photorealism, more recent work [6] has further shown that this can be avoided and extreme photorealism can be obtained through a hybrid setup. However, even so, a shortcoming of these models is that the components that constitute the scene—the geometry, appearance, and the lighting—are all *entangled* and are thus not controllable using user defined inputs.

Methods have been proposed to break this entanglement [2, 4, 35], however, they require multiview image collections of the scene being modeled and are thus inappli-

---

corresponding authors: {anuragr, otuzel}@apple.com

cable to images in-the-wild where such constraint cannot be satisfied easily. Boss et al. [3] loosens this constraint to images of different scenes, but they still require the same object to be seen from multiple views at the end. Moreover, these methods are not generative and therefore need to be trained for each object and cannot generate new objects. For generative methods [6, 7, 13], geometry and illumination remains entangled. In this work, we demonstrate that one does not require multiple views, and the variability and the volume of already existing datasets [16–18] are enough to learn a disentangled 3D generative model.

We propose *FaceLit*, a framework that learns a *disentangled* 3D model of a face, purely from images; see Fig. 1. The high-level idea behind our method is to build a rendering pipeline that is *forced* to respect physical lighting models [25,26], similar to [35] but in a framework friendly to 3D generative modeling, and one that can leverage off-the-shelf lighting and pose estimators [10]. In more detail, we embed the physics-based illumination model using Spherical Harmonics [26] within the recent generative Neural Volume Rendering pipeline, EG3D [6]. We then simply train for realism, and since the framework has to then obey physics to generate realistic images, it naturally learns a disentangled 3D generative model.

Importantly, the way we embed physics-based rendering into neural volume rendering is the core enabler of our method. As mentioned, to allow easy use of existing off-the-shelf illumination estimators [10], we base our method on Spherical Harmonics. We then model the diffuse and specular components of the scene via the Spherical Harmonic coefficients associated with the surface normals and the reflectance vectors, where the diffuse reflectance, the normal vectors and the material specular reflectance are generated by a neural network. While simple, our setup allows for effective disentanglement of illumination from the rendering process.

We show the effectiveness of our method using three datasets FFHQ [18], CelebA-HQ [16] and MetFaces [17] and obtain state-of-the-art FID scores among 3D aware generative models. Furthermore, to the best of our knowledge, our method is the very first generative method that can generate 3D faces with controllable scene lighting. Our code is available for research purposes at https://github.com/apple/ml-facelit/.

To summarize, our contributions are:

- we propose a novel framework that can learn a disentangled 3D generative model of faces from single views, with which we can render the face with different views and under various lighting conditions;

- we introduce how to embed an illumination model in the rendering framework that models the effects of diffuse and specular reflection.

- we show that our method can be trained *without* any man-

ual label and simply with 2D images and an off-the-shelf pose/illumination estimation method.

- we achieve state-of-the-art FID score of 3.5 among 3D GANs on the FFHQ dataset improving the recent work [6] by 25%, relatively.

2. Related work

We first discuss works that focus on decomposing a scene into shape, appearance, and lighting, then discuss those that focus on 3D generative models.

Decomposing a scene into shape, appearance and lighting. Neural Radiance Fields (NeRF) [21] learn a 5D radiance field where the aspects of the scene such as shape, appearance and lighting are jointly modeled by a neural network. While effective, this results in an entangled representation of the scene where only the camera pose can be controlled. In fact, besides controllability, in Ref-NeRF [34], it was demonstrated that explicitly allowing diffuse and specular rendering paths within the deep architecture, thus having in a sense architectural disentanglement leads to improved rendering quality. Thus, it is unsurprising that various methods have been proposed for disentanglement.

Recent works [2,35] use a Bidirectional Reflectance Distribution function (BRDF) to learn a scene representation using multiview images. Neural-PIL [4] computes a preintegrated lighting map from multiview image collections and models diffuse and specular reflectance in the scene. SAMURAI [3] operates on image collections of an object under different views and lighting conditions with various backgrounds thus reducing the strictness of the multiview constraint. However, even with reduced constraint, they still require the *same* object of interest to be in the scene from multiple views. In our case, we are interested in the problem setting when only single views are available, which none of these methods can be trivially extended to.

3D generative models. Generative Adversarial Networks (GAN) [12] trained from single-view images have been shown to be successful in generating photorealistic face images [18, 19]. Early works to enforce 3D consistency on generated images relied on conditioning the GAN with pose [22] and other attributes [11, 29] such as expression and lighting. However, these methods do not model the physical rendering process like NeRFs. This leads to inconsistent 3D geometry—rotating images generated from these 2D-based models result in change of shape and appearance and not just the view point. Follow up work [7,13,28,30,36] thus uses volume rendering—which brings 3D consistency by construction—on top of GANs to force 3D consistent representation. Furthermore, recent work [24, 27, 33] use illumination modeling on top of volume rendering. While this allowed the models to be 3D consistent, their generated image quality is not as photorealistic as their 2D counter-
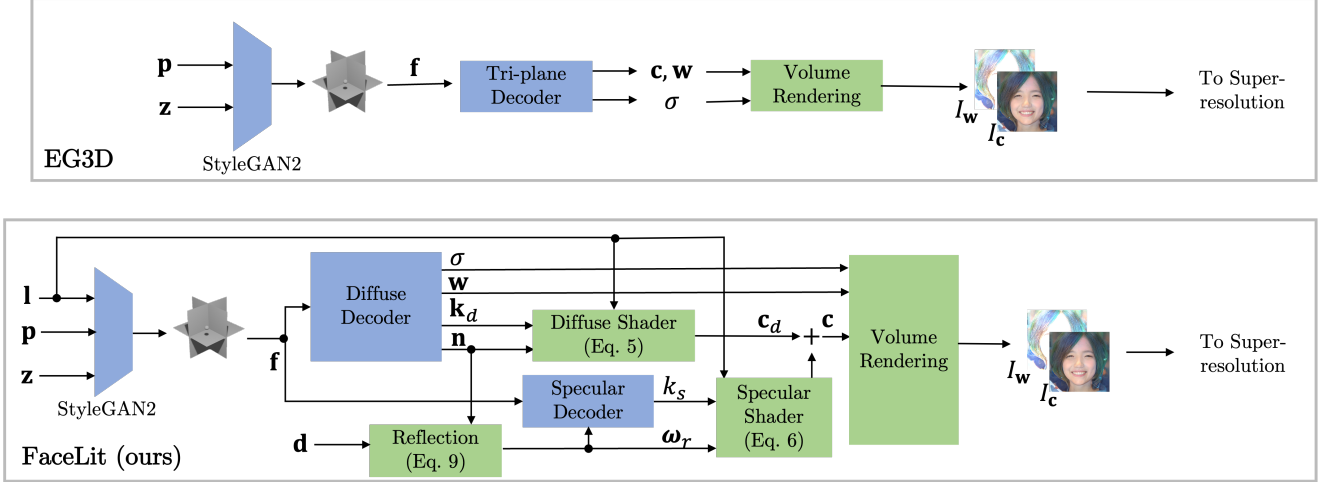
Figure 2. **Overview:** Our generation process samples a 3D face from a latent $\mathbf{z}$ conditioned on pose $\mathbf{p}$ and illumination $\mathbf{l}$. The generator uses a StyleGAN2 backbone with a tri-plane feature representation $\mathbf{f}$ introduced by EG3D [6] and introduces illumination modeling using diffuse and specular decoders. Green modules are non-trainable fixed differentiable functions. See §3.2 for details.

parts [19]. EG3D [6] thus proposes a tri-plane representation, and a hybrid framework that achieves the level of photorealism similar to 2D GAN frameworks. In a different direction, Cao et al. [5] utilize a massive dataset of multiple views of human faces to build a generative model that are conditioned on facial expressions, that are then adapted to a subject of interest for a controllable 3D model.

Regardless of whether these models aim for unconditional generation or controllability, they, however, do not disentangle geometry from illumination and thus cannot be relighted, limiting their application towards a fully controllable generative 3D model.

2.1. Preliminaries: The EG3D framework

As we base our framework on the EG3D framework [6], we first briefly explain the pipeline in more detail before discussing our method. The core of the EG3D pipeline is the use of tri-plane features, which allows the use of well-studied 2D CNNs for generating deep features to be used for volume rendering.

As shown in Fig. 2, EG3D uses a tri-plane generator $\mathcal{G}_{\text{tri}}$ with a StyleGAN2 [19] backbone conditioned on camera pose $\mathbf{p}$ to generate feature maps. These feature maps are re-arranged to obtain tri-plane features $\mathbf{f}^{XY}, \mathbf{f}^{YZ}, \mathbf{f}^{XZ}$ along 3 orthogonal planes. A decoder neural network is then used to regress color $\mathbf{c}$ and density σ and additional features $\mathbf{w}$ at a given location $\mathbf{x} \in \mathbb{R}^3$ from the tri-plane features. A color image $\mathbf{I_c}$ is then obtained by aggregating the values $\mathbf{c}, \sigma$ by volume rendering along the ray $\mathbf{r}$ given by

$$\mathbf{I_c}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \qquad (1)$$

where spatial locations are sampled within the near and far

plane locations as $t \in [t_n, t_f]$, $\mathbf{d}$ is the viewing direction, and the transmittance

$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(u))du\right). \qquad (2)$$

This volume rendering of $\mathbf{I_c}$ is performed at a relatively low resolution to be memory efficient, and an upsampling is performed for higher resolution images. Hence, an additional n_w-channel feature image $\mathbf{I_w}$ is rendered by tracing over the features $\mathbf{w} \in \mathbb{R}^{n_w}$, which is then used for generating the final image $\mathbf{I_c^+} = \mathcal{U}(\mathbf{I_c}, \mathbf{I_w})$, with the super-resolution network $\mathcal{U}$. The framework is then trained to make $\mathbf{I_c^+}$ as realistic as possible through GAN training setup, with the discriminator being conditioned on the poses $\mathbf{p}$.

Note here that, as the discriminator is conditioned on the pose, the pose must be provided for each training image; which has been shown to be effective in delivering better 3D consistency [6]. In the case of EG3D [6], these poses are obtained using Deng et al. [9]. Similarly, we will rely on DECA [10] since it provides the estimates of poses, as well as illumination.

3. Method

While modeling a scene with a radiance field is effective in rendering from novel viewpoints, it hinders our capability to relight the scene with different illumination conditions as illumination is entangled with the appearance and shape. To relight with unseen illumination conditions, we incorporate physics-based shading into the forming of radiance fields, thus disentangling it by construction.

In the following subsections we first discuss the illumination model that we propose which allows us to achieve

disentanglement, then detail how we implement the model into a deep generative model.

3.1. Illumination model

To explicitly constrain the rendering process on illumination we use a simplified version of the Phong reflectance model [25]. The color at a location is computed using

$$\mathbf{c} = \int_{\boldsymbol{\omega}} \left(\mathbf{k}_d \odot (\mathbf{n} \cdot \boldsymbol{\omega}) \, L^d(\boldsymbol{\omega}) + k_s \, (\boldsymbol{\omega}_r \cdot \boldsymbol{\omega})^\alpha \, L^s(\boldsymbol{\omega}) \right) \, d\boldsymbol{\omega},$$
(3)

where $\mathbf{n}$ is its normal, $\boldsymbol{\omega}_r$ is the reflection direction given by Eq. (9), $\mathbf{k}_d \in \mathbb{R}^3$ is the diffuse reflectance, $k_s \in \mathbb{R}$ is the specular reflectance coefficient, α is the shininess constant, and L^d and $L^s : \mathbb{R}^3 \to \mathbb{R}^3$ are the diffuse and specular environment maps (distance light distributions) respectively parameterized by incident light direction, $\boldsymbol{\omega}$ on the surface of the unit sphere. The operator $\odot$ is element-wise multiplication and $\cdot$ is the dot product. For brevity, when we element-wise multiply a scalar with a vector, we assume each element of the vector is multiplied by the scalar. Here, the first term computes the diffuse color $\mathbf{c}_d$ and the second term computes the specular color $\mathbf{c}_s$.

Further simplification via Spherical Harmonics. We assume a single environment map for diffuse and specular, $L^d = L^s = L$. To speed up rendering Eq. (3) and efficient representation, we follow Ramamoorthi et al. [26], and pre-integrate the environment map to compute the irradiance environment map

$$E(\mathbf{n}) = \int_{\boldsymbol{\omega}} (\mathbf{n} \cdot \boldsymbol{\omega}) \, L(\boldsymbol{\omega}) \, d\boldsymbol{\omega}.$$
(4)

The irradiance environment map can be efficiently (approximately) represented in Spherical Harmonics (SH) basis using only 9 basis functions. See [26] for details.

Consider that irradiance environment map is represented by SH coefficients $\mathbf{l}_k \in \mathbb{R}^3$ with SH basis $H_k : \mathbb{R}^3 \to \mathbb{R}$ and $k \in [1, 9]$. We can fold all illumination-related terms in Eq. (3) using the SH basis functions. Thus, the diffuse term can be rewritten as

$$\mathbf{c}_d = \mathbf{k}_d \odot \sum_k \mathbf{l}_k H_k(\mathbf{n}).$$
(5)

For the specular component we assume that $\alpha = 1$[1] in Eq. (3) which, with the SH basis again folding in the illumination terms gives us

$$\mathbf{c}_s = k_s \sum_k \mathbf{l}_k H_k(\boldsymbol{\omega}_r).$$
(6)

Note here that unlike in Eq. (5) we use $\boldsymbol{\omega}_r$ to retrieve the irradiance environment map values rather than $\mathbf{n}$, as we are

---

[1]Using $\alpha = 1$ allows us to use the same irradiance environment map to compute the specular color.

interested in the specular component, which reflects off the surface. The final color is then a composition of specular and diffuse components

$$\mathbf{c} = \mathbf{c}_d + \mathbf{c}_s$$
(7)

As we will show in Sec. 4, this simple formulation works surprisingly well, with the illumination being explicitly factored out. In other words, by controlling $\mathbf{l}_k$ in Eq. (5) and Eq. (6), one can control how the face renders under different illuminations. Although, we do not model other effects of light on the skin, such as subsurface scattering [20], we expect our model to account for it from the training process.

3.2. Generator

To imbue a 3D generative model with explicitly controllable illumination, we condition the tri-plane generator on both the camera pose $\mathbf{p}$ and the illumination $\mathbf{l}$. This allows us to take into account the distribution of illumination conditions within our training dataset, similar to how pose was considered in EG3D [6]. Mathematically, we write our tri-plane generator as

$$\mathbf{f}^{XY}, \mathbf{f}^{YZ}, \mathbf{f}^{XZ} = \mathcal{G}_{\text{tri}}(\mathbf{p}, \mathbf{l}).$$
(8)

For a given point in space $\mathbf{x} \in \mathbb{R}^3$, the aggregated features are obtained using $\mathbf{f_x} = \mathbf{f}_{\mathbf{x}}^{XY} + \mathbf{f}_{\mathbf{x}}^{YZ} + \mathbf{f}_{\mathbf{x}}^{XZ}$. However, as discussed earlier, simply conditioning the tri-plane generator $\mathcal{G}_{\text{tri}}$ alone is not enough to enable explicit and consistent control over the camera pose and illumination—there is no guarantee that the generated content will remain constant while illumination and camera pose changes. We thus utilize $\mathbf{f_x}$ and apply the illumination model in Sec. 3.1.

Specifically, as shown in Fig. 2, instead of directly regressing the color $\mathbf{c}$ and the density σ at a given point as in EG3D [6], we decode $\mathbf{f_x}$ using diffuse and specular decoders. We then apply shading through Eq. (5) and Eq. (6) to obtain the diffuse color $\mathbf{c}_d$ and the specular color $\mathbf{c}_s$ of a point.

Diffuse decoder. We regress the diffuse reflectance $\mathbf{k}_d$, normal $\mathbf{n}$ and density σ at a point given tri-plane features $\mathbf{f_x}$ using diffuse decoder (see Fig. 2). We then apply Eq. (5) to obtain the diffuse color $\mathbf{c}_d$. Here, as in other NeRF work [34, 35] we opt to directly regress the normals, as we also found that using the derivative of the density to be unreliable for training.

Specular decoder. We regress the specular reflectance coefficient k_s using the specular decoder. We also compute the reflection direction which is a function of the view direction $\mathbf{d}$ and the normal $\mathbf{n}$ given by

$$\boldsymbol{\omega}_r = \mathbf{d} - 2(\mathbf{d} \cdot \mathbf{n})\mathbf{n}.$$
(9)

We then obtain the specular color $\mathbf{c}_s$ using the specular shading model given by Eq. (6).

Volume rendering. Following [6, 23], we volume render the image $\mathbf{I_c}$ and feature images $\mathbf{I_w}$ by tracing over color $\mathbf{c}$ and features $\mathbf{w}$ respectively via Eq. (1).

Superresolution. Lastly, to generate high-resolution images, as in EG3D [6] we upsample the rendered images $\mathbf{I_c}$ to $\mathbf{I_c^+}$ via an upsampling module $\mathcal{U}$, guided by the feature images $\mathbf{I_w}$. We write

$$\mathbf{I_c^+} = \mathcal{U}(\mathbf{I_c}, \mathbf{I_w}). \tag{10}$$

3.3. Training

We extend the standard training process of EG3D [6] and adapt it to our framework. Specifically, for the GAN setup, we use the rendered image $\mathbf{I_c}$, and condition the discriminator on both the camera poses $\mathbf{p}$ and the illumination SH coefficients $\mathbf{l}$. We further introduce a regularization on the estimated normal $\mathbf{n}$, such that it matches the estimated densities. Similar to Zhang *et al.* [35], we introduce a loss term defined as

$$\mathcal{L}_\mathbf{n} = |\mathbf{n}(\mathbf{x}) - \nabla_\mathbf{x}\sigma(\mathbf{x})|_1, \tag{11}$$

where $\nabla_\mathbf{x}$ is the spatial gradient.

4. Experiments

4.1. Experimental setup

Datasets. We use three datasets for our experiments: FFHQ [18], MetFaces [17] and CelebA-HQ [16]. FFHQ contains 70,000 samples and CelebA-HQ contains 30,000 samples of real human faces as both datasets have been used traditionally to evaluate GAN methods. MetFaces contains 1,336 samples of faces taken from museum art images, which is a small dataset that we use to demonstrate that our method can be applied beyond real face photos.

Implementation details. As in Chan *et al.* [6], we mirror samples in each of the datasets to double the number of training samples. We estimate the camera poses $\mathbf{p}$ and illumination coefficients $\mathbf{l}$ using DECA [10].

We apply slightly varying training strategies for each dataset to account for their image resolution and volume. For FFHQ, we follow a strategy of EG3D [6] and train in two stages with a batch size of 32 on 8 GPUs. In the first stage, we train for 750k iterations where we volume render the images at 64^2 and super resolve them to 512^2. In the second stage, we adjust the rendering resolution to 128^2 and super resolve them to 512^2, and train them further for 750k iterations. For the CelebA-HQ dataset, we train only using the first stage at a rendering resolution of 64^2 that are superresolved to 512^2, for 500k iterations. For the MetFaces dataset, as the sample size is small, we use a model pretrained on FFHQ and fine tune it on MetFaces via ADA augmentation [17]. We train for 15k iterations. We detail the network architectures in the supplementary material.

4.2. Qualitative results

We first qualitatively demonstrate the effectiveness of our method.

Randomly drawn samples. We demonstrate the quality of our generated samples from the FFHQ dataset in Fig. 3. For our results, we also visualize the 3D reconstruction and the illumination used when generating these samples via a matt sphere for easy verification of the illumination consistency of each sample. As shown, our results are photorealistic, also with fine details. Furthermore, the 2D images rendered from our model is visually consistant with the 3D shapes. Note especially the regions around the lips and the teeth where our model provides improved 3D shape compared to EG3D, benefitting from the specular modeling.

We further show samples from CelebA-HQ and MetFaces datasets in Fig. 5. For both datasets our results deliver rendering quality that photorealistic, or indistinguishable form actual paintings, while still providing explicit control over illumination and camera pose. We provide additional visualizations in the supplementary material.

Controlling pose and illumination. In Fig. 4, we fix the latent code for generating a face and vary the camera pose and illumination. Each row corresponds to a different illumination condition. We note that the lighting matches over two different persons. We also see the effect of strong lights in the middle row versus weak directional lights on the top and the bottom row. Our model provides rendering that remains consistent regardless of the pose and illumination change. We show additional results in the supplementary material.

Specular highlights. In Fig. 6, we show the effect of varying the specular component on the generation process. We vary the strength of specular component across each row in Fig. 6. Note how the generated results for lower specularity seem matt, which is to be expected. Under normal conditions of specularity, highlights on the nose and cheeks are pronounced adding to the realism of the generated faces. Furthermore, using higher specularity results in glare.

4.3. Quantitative results

We further evaluate our results quantitatively through various metrics.

Evaluation metrics. We benchmark our generation quality on key metrics: FID score [15], KID score [1, 15], identity consistency (ID), depth, pose and light consistency.

- **FID [15] and KID [1, 15]:** As in [6], we sample 50,000 faces from the model and compute the score against the full dataset. The FID and KID score captures the photorealism in the generated samples.
- **Identity consistency (ID):** As in [6, 7, 31, 36], we compute ID with the mean Arcface [8] cosine similarity, after rendering a face with two random camera views. The

FaceLit (ours)   EG3D

Figure 3. **Qualitative results.** Curated generated faces with different pose and illumination conditioning using our model (left) compared with the curated generated samples from EG3D (right). Our model shows detailed reconstruction in the lip and teeth region. The environment map is rendered using the half-sphere at the bottom right.

measure highlights the consistency of the face under different rotations.

- **Consistency:** We further evaluate our model on camera pose, depth and illumination consistency. We sample 1024 faces from the generator along with their camera pose, illumination and depth using the our model. We then use DECA [10] to obtain psuedo ground truths for camera pose, illumination of the generated samples. This differs from baseline methods [6, 7, 31, 36] that use Deng et al. [9] for pose estimation and preprocessing. Since DECA has better performance, this also contributes to lower error in our evaluation. For depth consistency, following previous work [6, 31, 36], we estimate pseudo ground truth depth from Deng et al. [9]. We report the mean square error between our estimates and the pseudo ground truth estimates.

Variants. For the quantitative study we evaluate four different variants of our method. We introduce two sets of models FaceLit-d and FaceLit-f using the diffuse-only model and the full model respectively with volume rendering resolu-

tion of 64^2 and superresolved to 512^2. We further train the models FaceLit-D and FaceLit-F at volume rendering resolution of 128^2 and superresolved to 512^2.

FFHQ. We report the quatitative evaluations on the FFHQ dataset in Tab. 1 and compare with previous work on 3D aware GANs. We observe that FaceLit-D and FaceLit-F obtain state-of-the-art in photorealism metrics—FID, KID and competitive performance on ID and depth consistency metrics. Amongst methods that generate at 512^2 resolution, we achieve state-of-the-art accuracy on depth consistency metrics. We also note that although diffuse models achieve state-of-the-art on photorealism metrics, the full models that also model the specular components have better depth. On the pose consistency metrics, we achieve state-of-the-art performance.

MetFaces. In Tab. 2, we compare the performance of our model to a 2D GAN – StyleGAN2 [19], and a 3D aware GAN, StyleNeRF [13]. We obtain better photorealism than both the methods on FID and KID scores. We further provide face consistency metrics and 3D consistency metrics –
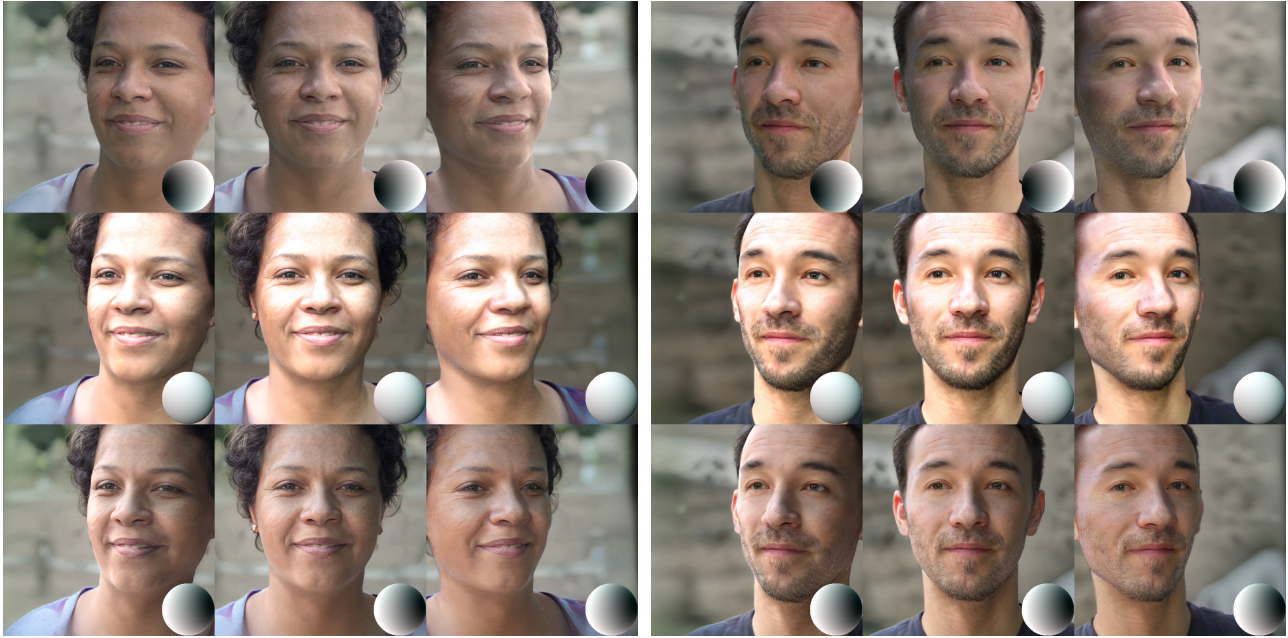
Figure 4. Camera Pose (left-to-right) and Illumination (top-to-bottom) interpolations. The illumination is represented by a normalized map [26] at the bottom-right of each image. Each row has same illumination. Top and bottom row show directional light. Middle row shows overexposed central light.

Figure 5. **Qualitative results.** Randomly generated faces with different pose and illumination conditioning using our model trained using MetFaces dataset (top) and CelebA-HQ dataset (bottom).

Figure 6. Effect of specular model by varying the specularity increasing from top to bottom. Specularity term helps in improving highlights on the nose and cheek region (middle row), and higher specularity results in the glaring effects (bottom row).

depth and pose. We note that the depth metrics obtained on this dataset are worse than those of FFHQ. A reason for this could be that art images do not strictly respect the physical model of illumination.

CelebA-HQ. In Tab. 3, we evaluate our models trained on a smaller dataset, CelebA-HQ with 30,000 samples. We note that our models can achieve better photorealism scores than 2D GANs such as StyleGAN and has good performance on consistency metrics. Furthermore, we also note that the full

	FID $\downarrow$	KID $\downarrow$	ID $\uparrow$	Depth $\downarrow$	Pose $\downarrow$
GIRAFFE [23] 256^2	31.5	1.992	0.64	0.94	0.089
π-GAN [7] 128^2	29.9	3.573	0.67	0.44	0.021
Lift. SG [31] 256^2	29.8	-	0.58	0.40	0.023
StyleNeRF [13] 512^2	7.80	0.220	-	-	-
StyleNeRF [13] 1024^2	8.10	0.240	-	-	-
GMPI [36] 512^2	8.29	0.454	0.74	0.46	0.006
GMPI [36] 1024^2	7.50	0.407	0.75	0.54	0.007
EG3D [6] 256^2	4.80	0.149	0.76	**0.31**	0.005
EG3D [6] 512^2	4.70	0.132	**0.77**	0.39	0.005
FaceLit-d 512^2	4.01	0.124	0.72	0.42	0.0009
FaceLit-f 512^2	4.06	0.115	0.72	0.33	**0.0008**
FaceLit-D 512^2	**3.48**	**0.097**	**0.77**	0.62	**0.0008**
FaceLit-F 512^2	3.90	0.117	0.75	0.43	**0.0008**

Table 1. Comparison of 3D aware GANs with FaceLit (ours) using FID, KID $\times 100$, Identity Consistency, depth accuracy and pose accuracy on the FFHQ dataset [18]. The method names are suffixed with the resolution of generated images.

	FID $\downarrow$	KID $\downarrow$	ID $\uparrow$	Depth $\downarrow$	Pose $\downarrow$
StyleGAN2 [19]	18.9	0.27	-	-	-
StyleNeRF [13]	20.4	0.33	-	-	-
FaceLit-d	**15.30**	**0.22**	**0.87**	0.96	**0.0018**
FaceLit-f	15.43	0.23	0.86	**0.77**	0.0032

Table 2. Comparison of a 2D GAN and a 3D aware GAN with FaceLit (ours) using FID, KID $\times$ 100, Identity Consistency, depth accuracy and pose accuracy on the MetFaces dataset [17]. All models generate at a resolution of 512^2.

	FID $\downarrow$	KID $\downarrow$	ID $\uparrow$	Depth $\downarrow$	Pose $\downarrow$
StyleGAN [18] 1024^2	4.41	-	-	-	-
FaceLit-d 512^2	**3.63**	**0.083**	**0.73**	0.36	**0.0011**
FaceLit-f 512^2	3.94	0.117	0.72	**0.33**	**0.0011**

Table 3. Performance of FaceLit evaluated using FID, KID $\times 100$, Identity Consistency, depth accuracy and pose accuracy on the CelebA-HQ dataset [16]

model, FaceLit-f provides better depth accuracy than diffuse only, FaceLit-d model.

4.4. Illumination accuracy

We further report how accurate our model learns the illumination effects by generating random samples and running DECA [10] on them to see how well the estimated illumination agree with the conditioned illumination. We use mean square error of the SH coefficients averaged over 1024 random samples from our model. The samples are conditioned on pose and illumination randomly sampled from the training dataset. While imperfect, as these results will be limited by the accuracy of DECA [10], it allows us to roughly gauge the accuracy of our illuminations. We report these results in Tab. 4. As reported, our model generates images that are well inline with the DECA estimates. We show the visualizations in Fig. 7.

	FFHQ	CelebA-HQ	MetFaces
FaceLit-d	0.0054	0.0061	0.0069
FaceLit-f	0.0053	0.0042	0.0084
FaceLit-D	0.0049	-	-
FaceLit-F	0.0051	-	-

Table 4. Average mean square error (MSE) of illumination SH coeffients evaluated using pseudo ground-truth from DECA [10]
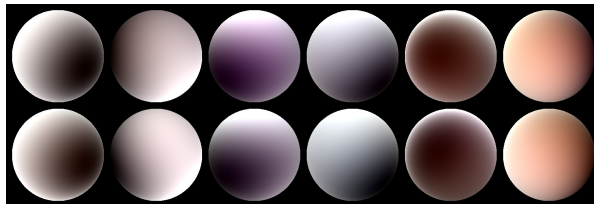

Figure 7. Top: Irradiance maps used to condition our generator. Bottom: Pseudo ground-truth estimates of irradiance maps using DECA [10] on the generated images.

5. Conclusion

We have presented a novel method to learn a disentangled 3D generative model for faces that allows the user to control the camera pose and the illumination, with only single views, and without the need of any manual annotation. Our core idea is how we model physics-based rendering with a simplified Phong model, that integrates effectively to neural volume rendering. We demonstrate the effectiveness of our method on FFHQ, CelebA-HQ, and MetFaces dataset, providing photorealistic image quality.

Limitations and future work. Our method does not model all the physical aspects of the scene and rendering process. Our setup is unsupervised, learned without any explicit control of the environment during the capture. While this problem setup is difficult, we are capable of generating photorealistic faces under various diffuse/specular lighting. The quality of the method may be further improved by modeling global illumination or subsurface scattering. For further improved factorization/specular quality, we may need higher frequency environment maps [14] that are difficult to obtain in an unsupervised setting where manual annotation or specially designed capture setup would help. Further, our method uses estimation of illumination parameters and camera pose from existing methods [10]. As such, the accuracy of our method is limited by the performance of existing work. In Fig. 3, it can be seen that the estimated illumination can sometimes be entangled with skin color, which is a limitation that we inherit from DECA [10].

Ethical Considerations. We intend our work to be used for research purposes only and should not be used edit images of real people without their consent. The use of our method for developing applications should carefully consider privacy of the faces, as well as, bias present in the datasets.

References

[1] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018. 5

[2] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 1, 2

[3] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T Barron, Hendrik Lensch, and Varun Jampani. Samurai: Shape and material from unconstrained real-world arbitrary image collections. *arXiv preprint arXiv:2205.15768*, 2022. 2

[4] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34:10691–10704, 2021. 1, 2

[5] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, Yaser Sheikh, and Jason Saragih. Authentic Volumetric Avatars from a Phone Scan. *ACM TOG*, 41(4), jul 2022. 3

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 8

[7] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021. 1, 2, 5, 6, 8

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 5

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3, 6

[10] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 2, 3, 5, 6, 8

[11] Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J Black, and Timo Bolkart. Gif: Generative interpretable faces. In *2020 International Conference on 3D Vision (3DV)*, pages 868–878. IEEE, 2020. 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[13] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022. 1, 2, 6, 8

[14] Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. Frequency domain normal map filtering. In *ACM SIGGRAPH 2007 papers*, pages 28–es. 2007. 8

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[16] 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*, 2018. 2, 5, 8

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 2, 5, 8

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2, 5, 8

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1, 2, 3, 6, 8

[20] Aravind Krishnaswamy and Gladimir VG Baranoski. A biophysically-based spectral model of light interaction with human skin. In *Computer graphics forum*, volume 23, pages 331–340. Wiley Online Library, 2004. 4

[21] 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*, pages 405–421. Springer, 2020. 1, 2

[22] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 2

[23] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 5, 8

[24] Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 34:20002–20013, 2021. 2

[25] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975. 2, 4

[26] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001. 1, 2, 4, 7

[27] Pramod Rao, Mallikarjun B R, Gereon Fox, Tim Weyrich, Bernd Bickel, Hans-Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Ayush Tewari, Christian Theobalt, and Mohamed Elgharib. Vorf: Volumetric relightable faces. 2022. 2

[28] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022. 1, 2

[29] Nataniel Ruiz, Barry-John Theobald, Anurag Ranjan, Ahmed Hussein Abdelaziz, and Nicholas Apostoloff. Morphgan: One-shot face synthesis gan for detecting recognition bias. *arXiv preprint arXiv:2012.05225*, 2020. 2

[30] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *arXiv preprint arXiv:2206.07695*, 2022. 2

[31] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2d stylegan for 3d-aware face generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6258–6266, 2021. 5, 6, 8

[32] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[33] Feitong Tan, Sean Fanello, Abhimitra Meka, Sergio Orts-Escolano, Danhang Tang, Rohit Pandey, Jonathan Taylor, Ping Tan, and Yinda Zhang. Volux-gan: A generative model for 3d face synthesis with hdri relighting. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2

[34] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. In *CVPR*, 2022. 2, 4

[35] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 1, 2, 4, 5

[36] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. *arXiv preprint arXiv:2207.10642*, 2022. 2, 5, 6, 8